

**VŠB - Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra měřicí a řídicí techniky**

**Vzdálené řízení a vizualizace procesů na SoftPLC  
WINCON 8000 s Windows CE .NET**

**Remote Control and Visualization of Processes on  
SoftPLC WINCON 8000 with Windows CE .NET**

**Ostrava, květen 2009**

**Michal Tutsch**

# Zadání diplomové práce

Student: **Bc. Michal Tutsch**

Studijní program: N2649 Elektrotechnika

Studijní obor: 2601T004 Měřicí a řídicí technika

Téma: **Vzdálené řízení a vizualizace procesů na SoftPLC WINCON 8000 s  
Windows CE .NET**  
**Remote Control and Visualization of Processes on SoftPLC WINCON  
8000 with Windows CE .NET**

Zásady pro vypracování:

Práce poskytne přehled nových možností platformy Windows CE .NET pro vzdálené řízení procesů. Bude obsahovat teoretický rozbor a následnou praktickou realizaci vybrané modelové úlohy pro vzdálené řízení a vizualizaci.

Práce bude obsahovat tyto části:

1. **Zpracování nových možností poskytovaných platformou Microsoft Windows CE .NET a nástroji Matlab Simulink a REX Control.**
2. **Výběr náročných modelových případů a konkrétní zpracování vybraného na modelovém příkladě (cílem je zjištění maximálních možností řízení se SoftPLC)**
3. **Stanovení limitů možností řízení a vizualizace pomocí SoftPLC platformy Windows CE .NET. Pro jaké maximální nároky kladené na rychlost řízení je řízení realizovatelné bez chyb?**
4. **Zhodnocení výsledků.**
5. **Extended abstract – shrnutí dosažených výsledků (4-6 stran v anglickém jazyce)**

Seznam doporučené odborné literatury:

- Kačmář, D. Programujeme .NET aplikace ve Visual Studiu .NET. ISBN 80-7226-569-5
- Andy Wigley, Stephen Wheelwright . Microsoft .NET Compact Framework. ISBN 0-7356-1725-2
- Lacko, I. Programujeme mobilní aplikace ve Visual Studiu .NET. ISBN 80-251-0176-2

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Ondřej Krejcar, Ph.D.**

Datum zadání: 30.11.2008

Datum odevzdání: 07.05.2009

---

doc. Ing. Jiří Koziorek, Ph.D.  
*vedoucí katedry*

---

prof. Ing. Ivo Vondrák, CSc.  
*děkan fakulty*

## **Prohlášení**

*Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně.  
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.*

.....  
*Michal Tutsch*

*Datum odevzdání diplomové práce :*

## **Poděkování**

Chtěl bych touto cestou poděkovat především vedoucímu mé diplomové práce panu Ing. Ondřeji Krejcarovi, Ph.D. za cenné rady, konzultace a připomínky spojené s vypracováním mého úkolu. Dále chci poděkovat své rodině, která mi vytváří klidné zázemí po celou dobu studia.

## **Abstrakt**

Diplomová práce se zabývá možnostmi vzdáleného řízení a vizualizací procesů na SoftPLC WinCon řady 8000 s Microsoft Windows CE .NET. Zaměřuje se především na využití řídicího systému REX společnosti REX Controls a jeho kompatibilitu s prostředím Matlab – Simulink pro měření a sběr dat. Dále je podrobně rozebráno připojení vizualizačních software jako např. Promotic k řídicímu systému REX pro vizualizaci na PC a následně vytvoření vizualizační aplikace v prostředí InduSoft Web Studio pro vizualizaci přímo z WinConu – na připojeném monitoru i přes webový server. Možnosti takto vytvořených řídicích i vizualizačních aplikací byly ověřeny na modelové úloze při řízení polohy pístnice hydromotoru.

## **Klíčová slova**

PAC, WinCon, REX, RDC, Simulink, InduSoft Web Studio

## **Abstract**

The diploma thesis describes possibilities of remote control and visualization of processes on SoftPLC WinCon 8000 based on platform Microsoft Windows CE .NET. It is concretely focused on usage of control system REX by company REX Controls and its compatibility with Matlab-Simulink for control and measurement of data. There is also in detail described design of connection with software for visualization like Promotic on PC as well as InduSoft Web Studio running straight in WinCon. Features and limits of control with WinCon were confirmed on model application when the position of hydraulic motor was controlled.

## **Keywords**

PAC, WinCon, REX, RDC, Simulink, InduSoft Web Studio

## Seznam použitých symbolů a zkratek

<b>API</b>	<b>Application Programming Interface.</b> Rozhraní pro programování aplikací.
<b>ASP</b>	<b>Active Server Pages.</b> Skriptovací prostředí pro tvorbu dynamických webových aplikací.
<b>CPU</b>	<b>Central Processing Unit.</b> Hlavní procesorová jednotka.
<b>DCS</b>	<b>Distributed Control System.</b> Distribuované systémy řízení.
<b>DHCP</b>	<b>Dynamic Host Configuration Protocol.</b> Protokol pro dynamickou konfiguraci ze serveru.
<b>DLL</b>	<b>Dynamic Linking Library.</b>
<b>EEPROM</b>	<b>Electrically Erasable Programmable Read-Only Memory.</b> Elektricky mazatelná semipermanentní paměť typu ROM-RAM.
<b>FBD</b>	<b>Function Block Diagram.</b> Jazyk funkčního blokového schématu
<b>FTP</b>	<b>File Transfer Protocol.</b> Umožňuje přenos souborů.
<b>HMI</b>	<b>Human - Machine Interface.</b> Operátorské rozhraní člověk – stroj.
<b>HTTP</b>	<b>HyperText Transport Protocol.</b> Protokol umožňující přenos dokumentů v různých formách.
<b>HVAC</b>	<b>Heating, Ventilation &amp; Air Conditioning.</b> Zkratka pro vytápění, klimatizace a vzduchotechniky.
<b>HW</b>	<b>Hardware.</b> Veškeré hmotné počítačové vybavení a součástky.
<b>I/O</b>	<b>Input/output.</b> Vstupně / výstupní.
<b>IL</b>	<b>Instruction List.</b> Jazyk seznamu instrukcí.
<b>IPC</b>	<b>Industrial Personal Computer.</b> Průmyslový počítač.
<b>IWS</b>	<b>Indusoft Web Studio.</b> Vývojové prostředí vizualizačních aplikací pro WinCon.
<b>LD</b>	<b>Ladder Diagram.</b> Jazyk kontaktních schémat.
<b>MAC</b>	<b>Media Access Control.</b> Adresa jednoznačně identifikující síťové zařízení.
<b>ODBC</b>	<b>Open Database Connectivity.</b> Jedná se o standardní aplikační rozhraní (tzv. API) pro přístup k datům. Použitím API ODBC mohou aplikace přistupovat k datům, nezávisle na tom, jakým systémem pro řízení báze dat jsou tato data spravována.
<b>OPC</b>	<b>OLE for Process Control.</b> Komunikační rozhraní mezi SCADA sys. a PLC.
<b>PAC</b>	<b>Programmable Automation Controller.</b> Programovatelné řídicí automaty.
<b>PC</b>	<b>Personal Computer.</b> Osobní počítač.
<b>PDA</b>	<b>Personal Digital Asistent.</b> Osobní digitální asistent, neboli PDA.
<b>PLC</b>	<b>Programmable Logic Controller.</b> Programovatelný logický automat.
<b>RISC</b>	<b>Reduced Instruction Set Computer.</b> Počítač s redukovanou instrukční sadou.
<b>SCADA</b>	<b>Supervisory Control And Data Acquisition.</b> Sběr dat a řízení technologických procesů.
<b>SDK</b>	<b>Software Development Kit.</b> Sada nástrojů pro vývojáře
<b>SDRAM</b>	<b>Synchronous Dynamic Random Access Memory.</b> Paměť s náhodným přístupem.
<b>SFC</b>	<b>Sequential Function Chart.</b> Sekvenční funkční diagram.
<b>SOAP</b>	<b>Simple Object Access Protocol.</b> Typ komunikace mezi aplikacemi na bázi webových služeb (web services).
<b>SQL</b>	<b>Structured Query Language.</b> Strukturovaný dotazovací jazyk.
<b>ST</b>	<b>Structured Text.</b> Jazyk strukturovaného textu.



<b>TCP/IP</b>	<b>Transmission Control Protocol / Internet Protocol.</b> Sada síťových protokolů používaných v síti Internet, která poskytuje komunikaci v rámci vzájemně propojených sítí tvořených počítači s různou hardwarovou architekturou a různými operačními systémy.
<b>TELNET</b>	<b>Telecommunication Network.</b> Poskytuje službu vzdáleného terminálového přístupu atd...
<b>UDP</b>	<b>User Datagram Protocol.</b> Sada protokolů transportní vrstvy používaný v síti ethernet.
<b>USB</b>	<b>Universal Serial Bus.</b> Univerzální sériová sběrnice.
<b>WiFi</b>	<b>Wireless Fidelity.</b> Bezdrátová síť.

# Obsah

<b>1. ÚVOD .....</b>	<b>1</b>
<b>2. ZÁKLADNÍ POPIS PAC WINCON ŘADY 8000 .....</b>	<b>2</b>
2.1 TYPOVÉ OZNAČENÍ PAC WINCON ŘADY 8000 .....	3
2.2 POPIS HLAVNÍ JEDNOTKY PAC WINCON ŘADY 8X3X.....	4
2.3 DOSTUPNÁ HARDWAROVÁ KONFIGURACE WINCONŮ .....	6
2.4 SYSTÉMOVÉ NÁSTROJE WINCONU V PROSTŘEDÍ MS WINDOWS CE.....	7
2.5 DALŠÍ DOSTUPNÉ TYPY PAC VÝROBCE ICPDAS .....	11
<b>3. MOŽNOSTI TVORBY ŘÍDICÍCH A VIZUALIZAČNÍCH APLIKACÍ.....</b>	<b>13</b>
3.1 SDK PRO WINCON NA PLATFORMĚ .NET FRAMEWORK.....	13
3.2 IPUSH EMBEDDED.....	14
3.3 MATLAB – SIMULINK.....	15
3.4 ŘÍDICÍ SYSTÉM REX .....	16
3.4.1 <i>Současné podporované platformy ŘS REX:</i> .....	16
3.4.2 <i>Struktura řídicího systému REX</i> .....	16
3.4.3 <i>REX + WinCon 8x31 (Windows CE)</i> .....	17
3.4.4 <i>Řídicí systém REX a vstupně-výstupní ovladače</i> .....	18
3.4.5 <i>Knihovna funkčních bloků RexLib:</i> .....	19
3.4.6 <i>Příklad konfigurace</i> .....	23
3.4.7 <i>OPC Server v instalaci REXu</i> .....	24
3.4.8 <i>Trendy, archivace a přenos procesních dat REXu do Matlabu :</i> .....	25
3.5 INDUSOFT WEB STUDIO.....	26
3.6 VIRTUAL CE 4.2 PRO – VZDÁLENÁ PLOCHA .....	26
<b>4. MODELOVÉ ÚLOHY VZDÁLENÉHO ŘÍZENÍ A VIZUALIZACE.....</b>	<b>27</b>
4.1 VÝBĚR VHODNÝCH MODELOVÝCH PŘÍPADŮ PRO SYSTÉM WINCON + REX .....	27
4.2 POPIS ZVOLENÉ ŘÍZENÉ TECHNOLOGIE .....	28
4.3 PROPOJENÍ ŘÍDICÍ APLIKACE V REXU SE SCHÉMATEM V SIMULINKU .....	30
4.3.1 <i>Řízení z Matlab-Simulinku</i> .....	30
4.3.2 <i>Řízení z REXu ve WinConu a zobrazování průběhů v Matlabu</i> .....	32
4.4 ŘÍDICÍ SCHÉMA REXU PRO PŘIPOJENÍ RŮZNÝCH VIZUALIZAČNÍCH APLIKACÍ .....	35
4.4.1 <i>Návrh řídicího schématu reálné úlohy</i> .....	35
4.4.2 <i>Návrh simulačního řídicího schématu pro model soustavy</i> .....	37
4.4.3 <i>Diagnostika schématu prováděného exekutivou REXu</i> .....	39
4.5 VIZUALIZAČNÍ APLIKACE V PROSTŘEDÍ PROMOTIC.....	41
4.6 VIZUALIZAČNÍ APLIKACE V PROSTŘEDÍ INDUSOFT WEB STUDIO .....	43
4.7 HISTORICKÉ TRENDY VYČÍTANÉ Z APLIKACE REXVIEW .....	46
4.8 MĚŘENÍ REÁLNÉ PERIODY SPOUŠTĚNÍ ŘÍDICÍHO SCHÉMATU REXU .....	48
4.9 ZHODNOCENÍ DOSAŽENÝCH VÝSLEDKŮ .....	50
<b>5. ZÁVĚR .....</b>	<b>53</b>
<b>6. EXTENDED ABSTRACT .....</b>	<b>54</b>
<b>7. POUŽITÁ LITERATURA .....</b>	<b>58</b>
<b>8. SEZNAM PŘÍLOH .....</b>	<b>59</b>

## 1. Úvod

Cílem této diplomové práce je poskytnout přehled možností vzdáleného řízení SoftPLC WinCon řady 8000 a definovat limity řízení při využití řídicího software REX firmy REX Controls a Matlab-Simulinkem pro měření a sběr dat. Dále popsat propojení s vizualizačním software na PC jako např. Promotic i s vizualizačním software určeným přímo pro platformu Windows CE jako je Indusoft WebStudio. Definovat základní limity řízení při použití již zmíněných řídicích systémů a následný návrh a realizace demonstrační úlohy pro řízení modelové úlohy s cílem zjištění maximálních možností řízení se SoftPLC. Diplomová práce navazuje na bakalářskou práci [1], kde bylo řízení i následná vzdálená vizualizace procesů řešena vývojem aplikací v Microsoft Visual Studiu.

Text je rozdělen do osmi kapitol. První kapitolou je úvod. Druhá kapitola je věnována popisu SoftPLC WinCon řady 8000, jeho vlastnostem i komunikačním schopnostem, možnostem připojení rozšiřujících I/O modulů a popis konkrétní použité konfigurace. Dále jsou zde uvedeny ostatní příbuzné modely PAC výrobce ICPDAS a jejich stručný popis. Třetí kapitola rozebírá softwarové vybavení PAC WinCon a možnosti vzdáleného řízení a vizualizace v podobě různých vývojových nástrojů, přičemž podrobněji je zde popis řídicího systému REX. Čtvrtá kapitola podrobně popisuje návrh modelové úlohy pro vzdálené řízení a vizualizaci. Součástí kapitoly je návrh několika typů řídicích algoritmů v REXu, vizualizační aplikace v Promotic a InduSoft Web Studio a následné shrnutí všech dosažených výsledků a zjištěných poznatků. Pátá kapitola je závěr. Šestá kapitola je tzv. „Extended abstract“, tedy shrnutí dosažených výsledků v anglickém jazyce. Následují kapitoly seznam odkazů na použitou literaturu a seznam příloh.

## 2. Základní popis PAC WinCon řady 8000

Následující kapitola popisuje základní vlastnosti a zařazení programovatelného řídicího kontroléru ICPDAS WinCon řady 8000 jako průmyslový modulární systém PAC.

PAC je zkratka vytvořená společností Automation Research Corporation (ARC) znamenající programovatelný řídicí kontrolér (Programmable Automation Controller). Označuje se jím nová generace průmyslových řídicích prvků kombinujících spolehlivost a modularitu PLC a zároveň flexibilitu s výkonem technologie dnešních PC. Tyto zařízení jsou tedy schopny nejen řídit zvolený technologický proces (jako samostatné PLC), ale zároveň provádět například vizualizaci průběhů zvolených technologických veličin, jejich záznam do vlastních databází případně přímo distribuovat tyto data standardizovanými komunikačními formáty jako jsou OPC servery, DDE servery apod.. [1]

PAC WinCon je založený na CPU Intel Strong ARM RISC s dostatečnou kapacitou operační paměti. Systém je rozšiřitelný ucelenou řadou vstupních a výstupních modulů se zásuvným systémem typickým pro automaty PLC a připojitelných přímo k základní nebo rozšiřující jednotce. Nabídka zahrnuje jak analogové tak digitální vstupy a výstupy, frekvenční čítače, moduly pro řízení pohonů, rozšiřující sériová rozhraní apod. Rozšiřující jednotky je možné připojit přes sériovou nebo ethernetové rozhraní. Součástí každé základní jednotky jsou dále rozhraní USB, PS/2, VGA, ethernet, sériové porty a slot pro paměťovou kartu Compact Flash. K těmto rozhraním lze připojit celou řadu standardních PC-periferií jako VGA displeje, USB disky, klávesnice a myši, dotykové displeje apod. Systém je navržen pro provoz bet ventilátorů a mechanických disků, má relativně nízkou spotřebu a samozřejmě je schopný provozu v průmyslu při velkém rozsahu teplot. [1]

Jako základ programového vybavení je použit deterministický operační systém reálného času Windows CE.NET 4.1, který je ve WinCon uložený v paměti typu flash. Díky tomu se dosahuje rychlého startu celého systému, předvídatelnosti jeho chování i snadného uvedení systému do výchozího stavu. Samotný obraz operačního systému lze jednoduše přehrávat novějšími verzemi, které jsou volně dostupné na webu výrobce. Jednotlivé verze obrazů operačního systému se liší i některými prvky, jakou jsou třeba typy podporovaných dotykových panelů. Výhodou operačního systému Windows CE .NET je programové rozhraní .NET Compact Framework verze 1.0 SP3, které podstatně zjednodušuje vývojářům přechod na platformu WinCon například od aplikací pro stolní PC s operačními systémy MS Windows.

Platforma WinConu zároveň umožňuje jeho použití HMI/SCADA systém, s vizualizací řízeného procesu včetně možnosti ovládání procesu přes již zmíněný dotykový panel. Z důvodu podpory veškerého hardware nabízeného firmou ICPDAS je jako nejvhodnější SCADA software doporučováno systém InduSoft Web Studio, jehož runtime pro operační systém Windows CE.NET může být dodán už jako před-instalovaná součást PAC systémů WinCon. Tímto systémem lze navrhnout a nahrát do WinConu vizualizační aplikaci zobrazující stav technologického procesu jak na připojeném monitoru, tak přes webový server umístěný přímo ve WinConu kdy jako vzdálený klient poslouží libovolný webový prohlížeč. [3]

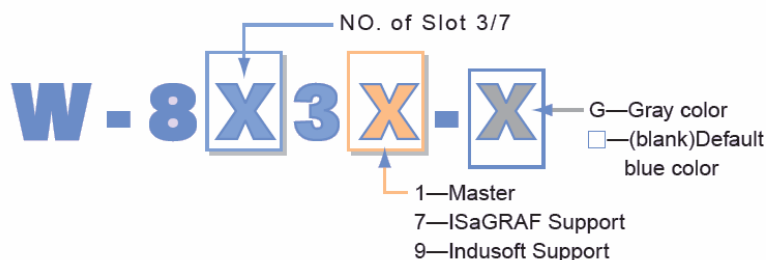
Klasické známé prostředí pro programování automatů PLC je programátorům zajištěno také, protože software pro WinCon-8000 může být vyvíjen i v systému InduSoft ISaGRAF, který podporuje všech pět jazyků definovaných normou IEC61131-3. [4]

WinCon série 8000 je kompatibilní s balíkem pro podporu vývoje aplikací WinCon SDK, včetně OPC serveru, Modbus knihoven, funkcí pro práci se zásuvnými a distribuovanými moduly. Součástí operačního systému je také MS SQL server CE 2.0 pro databázové aplikace. Možnosti aplikací pro PAC systémy WinCon nejsou předem omezeny. Aplikace mohou provádět nejen řízení a zpracování dat, ale i jejich vizualizaci a komunikaci s nadřazenými systémy. K dispozici pro vývoj aplikací jsou komerční vývojová prostředí Embedded Visual C++ a Visual Studio .NET.

V nedávné době byl firmou REX Controls vyvinut řídicí systém REX, jehož jednou z možných cílových platform je i WinCon. Řídicí systém REX je kompatibilní s Matlab-Simulinkem. To umožňuje vytvářet řídicí algoritmy s využitím bloků řídicího systému REX přímo v Matlab-Simulinku a odtud je přímo simulovat a následně i nahrávat do WinConu. Pro plnohodnotné využití systému REX však licence Simulinku není nutnou podmínkou, jelikož systém REX v sobě již obsahuje plnohodnotné vývojové prostředí, které však neumožňuje přímo simulovat chování řídicích algoritmů pro REX. Připojení vizualizačních aplikací k REXu je možné přes vestavěný OPC Server, který existuje jak pro platformu REXu tak různých operačních systémů na PC. [5]

## 2.1 Typové označení PAC WinCon řady 8000

V současné době existují dva lehce odlišné typy základních jednotek WinCon. Jednak je to řada WinCon-8x3x. Ta existuje v provedení pro žádný, tři nebo sedm paralelních nebo sériových zásuvných modulů. Hlavní procesorová jednotka je zde Intel Strong ARM taktovaných na 206MHz, dále 32MB flash paměť a operační paměť 64MB SDRAM. Vždy obsahuje slot pro Compact Flash kartu, rozhraní ethernet, PS/2, VGA a USB port a sériový port RS-232 a RS-485. Základní jednotka může být dodána i s před-instalovanou licencí pro Indusoft IsaGraf nebo Indusoft Web Studio. Značení jednotlivých typů WinConu řady 8x3x ilustruje schéma na následujícím obrázku :



Obr. 1: Značení jednotlivých modelů řady WinCon 8000

Přinstalovaný software W-8x3x	Jednotlivé modely
Standard WinCon pouze s SDK	W-8031/8331/8731
ISaGRAF Software	W-8037/8337/8737
InduSoft Software	W-8039/8339/8739

Tabulka 1: Jednotlivé typy PAC WinCon řady 8x3x

Dalším typem je novější řada WinCon-8x4x, které je odvozena od základní řady WinConů 8x3x. Používá i stejný operační systém. Oproti základní řadě mají tyto WinCony doplněny druhý USB port pro připojení většího množství periférií, druhé Ethernet rozhraní a otočný přepínač pro konfiguraci systému a aplikací. Ze systému byly vypuštěny PS/2 porty. Dvojitě Ethernet rozhraní poskytuje větší propustnost sítě, možnost vyššího zabezpečení a redundance. Umožňuje vytvořit záložní síťovou cestu, nebo použít jedno rozhraní pro připojení do podnikové sítě a druhé pro připojení dalších řídicích systémů nebo senzorů průmyslovým Ethernetem, což zvyšuje celkovou bezpečnost a spolehlivost systému. [3]

Přinstalovaný software W-8x4x	Jednotlivé modely
Standard WinCon pouze s SDK	W-8041/8341/8741
ISaGRAF Software	W-8047/8347/8747
InduSoft Software	W-8049/8349/8749

**Tabulka 2: Jednotlivé typy PAC WinCon řady 8x4x**

## 2.2 Popis hlavní jednotky PAC Wincon řady 8x3x

Na levé části systému PAC WinCon 8x3x (viz Obr. 2) se nachází CPU modul, který osazený standardním 15 pinovým VGA portem pro připojení externího monitoru případně dotykového panelu, jedním USB 1.1 portem a ethernetovým portem (10 Base T). Na pravé části je sběrnice pro zásuvné I/O moduly ICPDAS série I-8000 a I-87K se třemi případně sedmi volnými sloty, existují však i modely bez rozšiřující sběrnice. Na boční straně systému jsou pak sériové porty COM2 s rozhraním RS 232 a 9 pinovým D-Sub konektorem pro připojení standardních sériových zařízení na vzdálenost až 15m. Následně COM3 s rozhraním RS-485 a šroubovací svorkovnicí sloužící primárně pro připojení vzdálených rozšiřujících jednotek se zásuvnými I/O moduly. Každá rozšiřující jednotka obsahuje vlastní napájecí zdroj a 4, 5, 8 nebo 9 sériových slotů pro zásuvné rozšiřující moduly v externí sériové sběrnici (konkrétně modely 87K4 až 87K9). Jedna řídicí jednotka však umožňuje ovládání maximálně 255 modulů přes jeden port RS-485.

Existují dva typy zásuvných I/O modulů, paralelní a sériové. Oba se připojují stejným typem konektoru, ale jejich komunikace s hlavním CPU je značně rozdílná. Paralelní vysokorychlostní moduly řady I-8000 lze umístit pouze do hlavní řídicí jednotky a s hlavním CPU komunikují po vlastní vnitřní sběrnici. Sériové I/O moduly řady 87K mohou být instalovány jak do paralelních zásuvných slotů hlavní jednotky tak do sériových zásuvných slotů rozšiřujících jednotek. Jejich nevýhodou je však nižší rychlost přenosu dat, jelikož komunikují s hlavním CPU přes sériové rozhraní. [3]



Obr. 2: Čelní pohled na PAC WinCon W-8739 se zapojenými čtyřmi měřicími moduly

<b>procesor</b>	Intel Strong ARM CPU, 206MHz
<b>operační systém</b>	Windows CE.NET 4.2
<b>SDRAM</b>	64M bytes
<b>Flash paměť</b>	32M bytes
<b>EEPROM:</b>	16K bytes
<b>Paměťová karta</b>	Compact Flash
<b>Ethernetový port</b>	10 Base T (RJ45)
<b>VGA Port</b>	Podporuje rozlišení od 320x240 až po 1024x768
<b>2 PS/2 porty</b>	klávesnice a myš
<b>1 USB 1.1 Host</b>	USB disk nebo USB myš
<b>COM Port</b>	9-pinový D-Sub (RS-232) a dvoudrátový šroubovací RS-485
Watchdog Timer	
Hodiny reálného času	
<b>Provozní teplota</b>	-25°C až +75°C
<b>Skladovací teplota</b>	-30°C až +85°C
<b>Provozní vlhkost vzduchu</b>	5~95%
<b>Napájení</b>	25W, od +10Vdc do +30Vdc
<b>Rozměry(ŠxVxH)(mm)</b>	114x110x90(0 rozšiřujících slotů) 229x110x90(3 rozšiřující sloty) 354x110x90(7 rozšiřujících slotů)

Tabulka 3: Technické parametry SoftPLC WinCon řady 8000

## 2.3 Dostupná hardwarová konfigurace WinConů

Pro vývoj, testování a ladění aplikací byl k dispozici WinCon 8331 se třemi volnými sloty a pěti zásuvnými moduly:

- I-8017 – modul obsahující 8 analogových vstupů se 14 bitovým A/D převodem, napěťovými rozsahy  $\pm 10V$ ,  $\pm 5V$ ,  $\pm 2.5V$ ,  $\pm 1.25V$ ,  $\pm 20mA$  (s externím rezistorem  $125\Omega$ )
- I-8024 – 4 analogové výstupy se 14 bitovým D/A převodem, napěťový rozsah  $\pm 10V$
- I-8054 – modul obsahující 8 digitálních vstupů a 8 digitálních výstupů, I-8056 – modul obsahující 16 digitálních výstupů
- I-87024 - modul obsahující 4 analogové výstupy se 14 bitovým D/A převodem, napěťový rozsah  $\pm 10V$ , možnost umístění i v rozšiřující jednotce

Pro nasazení do reálné modelové úlohy při řízení polohy pístnice hydromotoru byl využíván model PAC WinConu 8731 se sedmi volnými sloty a čtyřmi zásuvnými moduly: I-8017 (dvakrát), I-8054 (tento modul nebyl v úloze využit), I-87024.

Input Channels	8-ch Differential
Input Range	$\pm 10V$ , $\pm 5V$ , $\pm 2.5V$ , $\pm 1.25V$ $\pm 20mA \sim \pm 20mA$ (Requires Optional External 125 Ohm Resistor)
Resolution	14-bit
Sample Rate	Single Channel Polling Mode : 100K S/s Single Channel Interrupt Mode: 50K S/s 8 channel Scan Mode: 16 K S/s
Accuracy	$\pm 0.1\%$ of FSR
Input Impedance	20K, 200K, 20M (Jumper Select)
Input Bandwidth	100 KHz

**Tabulka 4: Parametry modulu I-8017**

Output Channels	4
Output Type	$0 \sim +5V$ , $\pm 5V$ , $0 \sim +10V$ , $\pm 10V$ , $0 \sim +20mA$ , $+4 \sim +20mA$
Resolution	14-bit
Accuracy	$\pm 0.1\%$ of FSR
Readback Accuracy	$\pm 1\%$ of FSR
Zero Drift	Voltage: $\pm 30\mu V/^\circ C$ Current: $\pm 0.2\mu A/^\circ C$
Span Drift	$\pm 20ppm/^\circ C$
Programmable Output Slope	0.125 to 2048 mA/ second 0.0625 to 1024 V/ second
Voltage Output Capability	10V@5mA
Current Load Resistance	External +24V : 1050 Ohms
Power-Up and SafeValue	Yes

**Tabulka 5: Parametry modulu I-87024**

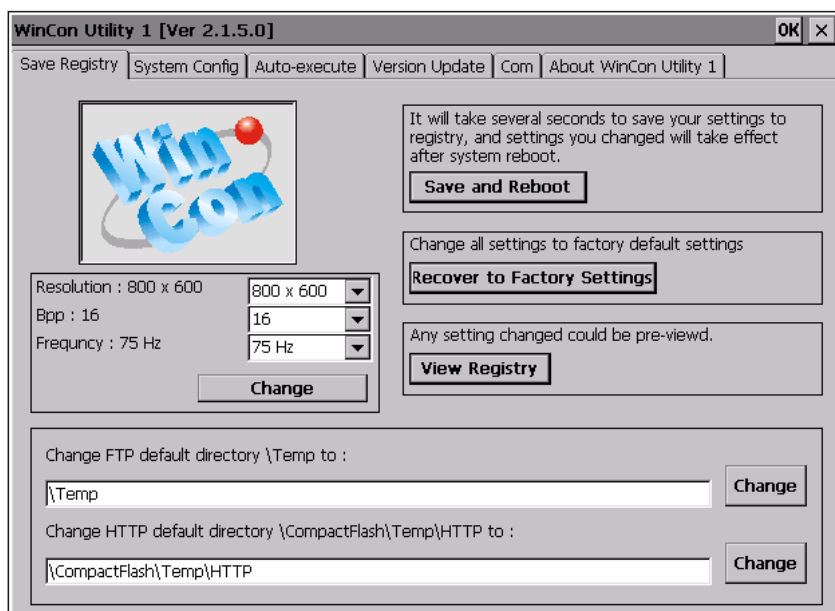


## 2.4 Systémové nástroje WinConu v prostředí MS Windows CE

Při prvním startu WinConu je třeba, stejně jako u jiných zařízení s operačními systémy Microsoft Windows CE, provést základní nastavení některých systémových prostředků. Jedná se zejména o nastavení času a časového pásma, parametrů ethernetové sítě, apod. Toto prvotní nastavení vyžaduje připojený monitor a alespoň počítačovou myš na portu PS/2, klávesnice není nutnou podmínkou, jelikož lze zapnout a používat tzv. „softwarovou“ klávesnici. Po připojení všech periférií a napojení zařízení WinCon do napájecí sítě automaticky naběhne operační systém a všechny jeho podprogramy nutné pro obsluhu externích periférií i zásuvných modulů. Na monitoru se objeví grafické prostředí Windows CE s hlavním panelem a tlačítkem „Start“.

V první řadě je nyní vhodné nastavit správný systémový čas. To je podobné jako u PC s operačním systémem Windows, tedy z nabídky „Start -> Settings -> Control panel -> Date/Time“. Dále je třeba nastavit parametry připojení do ethernetové sítě. Opět se přes nabídku „Start -> Settings -> Network and Dial-up Connections“ zobrazí okno s nastavením jednotlivých parametrů. Přidělování IP adresy je standardně na základě DHCP serveru nebo definováním statické adresy, masky podsítě a výchozí brány. Posledním krokem je nastavení identifikace zařízení v rámci sítě, k tomu se lze dostat opět přes nabídku „Start -> Settings -> Control panel -> System“. Aby se všechny tyto změny zapsaly do stálé paměti a uchovaly se i po restartu zařízení, je nutné uložit je do registrů pomocí aplikace „WinCon Utility“ z nabídky start.

Aplikace „WinCon Utility“ (viz Obr. 3) má primárně funkci ukládání systémových informací do pevné paměti a dále umožňuje některé nastavení WinConu. Mezi tyto na prvním panelu patří rozlišení monitoru (320x240 až 1024x768 pixelů), obnovovací frekvence a počet zobrazovaných barev (2,4,8 nebo 16 bitů). Dále je zde udávána cesta k adresáři sloužící pro FTP přístup a cesta k adresáři pro zobrazení stránek http serveru. Tlačítkem „Save and Reboot“ se uloží všechny systémové změny



Obr. 3: Grafické prostředí aplikace WinCon Utility

Čtvrtý panel slouží pro aktualizace obrazu operačního systému, pátý pro výběr typu případného dotykového panelu na COM portu a vyhledání a zobrazení vzdáleně připojených rozšiřujících jednotek

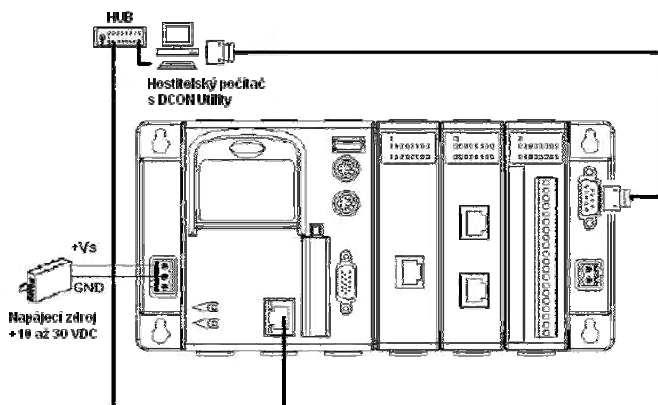
a provede restart systému. Na následujícím panelu se zobrazují zásuvné moduly v jednotlivých slotech a informace o verzi obrazu nainstalovaného operačního systému, MAC adrese síťové karty, paměti EEPROM, FLASH, a verzi SDK. Na třetím panelu „Auto-execute“ lze nastavit cestu k programům, které se mají automaticky spustit, jakmile naběhne operační systém. Zde by měli být definovány programy jako řídicí systém REX nebo Indusoft Web Studio.

se zásuvnými moduly. Po každém zapojení nového modulu nebo jejich výměně je nutné tuto proceduru spustit.

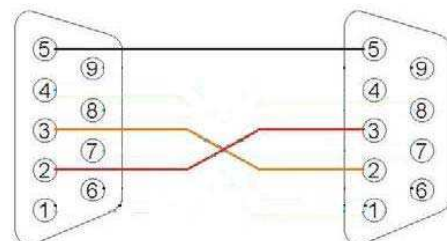
Jak již bylo psáno dříve, existují dva typy zásuvných modulů – paralelní (I-8K) a sériové (I-87K). Paralelní moduly lze umístit pouze do slotů hlavní jednotky WinCon, přičemž není nutná jejich zvláštní inicializace. Všechny nastavení těchto modulů lze řešit přímo programově. Sériové zásuvné moduly je možné umístit jak do slotů hlavní jednotky tak do rozšiřujících jednotek, přičemž v obou případech je nutné provést inicializační proces. Ten obnáší nastavení adresy zásuvného modulu, jejich rychlost (bitrate) komunikace a především nastavení parametrů vstupních a výstupních obvodů (např. výběr napěťového nebo proudového výstupu, rozsahy výstupů, vstupů apod). Inicializaci lze provádět dvěma způsoby. Buďto přímo z WinConu, nebo komfortněji z PC.

První možností je spustit přímo na WinConu utilitu *I87k\_In\_Slot.exe*, která je k dispozici na paměťové kartě v adresáři */CompactFlash/ICPDAS/Tools* na WinConu. Poté zadáváním přesně definované sady příkazů výrobcem ICPDAS a to do zadávacího prvku „Command“ provádět inicializaci a nastavení modulů. Tento způsob lze použít pouze v případě, že konfiguruje moduly zapojené přímo do hlavní jednotky WinCon, pro moduly v rozšiřující vaně tohle nelze.

Druhou možností inicializace sériových modulů je využít program *DCON\_Utility*. Ten je možné volně stáhnout z webu ICPDAS. Program je určen pro nastavení sériových modulů jak zasunutých přímo do WinConu tak i do rozšiřující vany. Inicializace se pak provádí jednodušeji a to na PC s běžícím prostředím *DCON\_Utility*, přičemž PC musí být připojeno k WinConu přes Ethernet a zároveň sériovým kabelem (viz Obr. 5 a Obr. 4).



Obr. 5: Schéma připojení PC a WinConu při prvotní inicializaci modulů přes aplikace DCON\_Utility



Obr. 4: Propojení jednotlivých pinů sériového kabelu

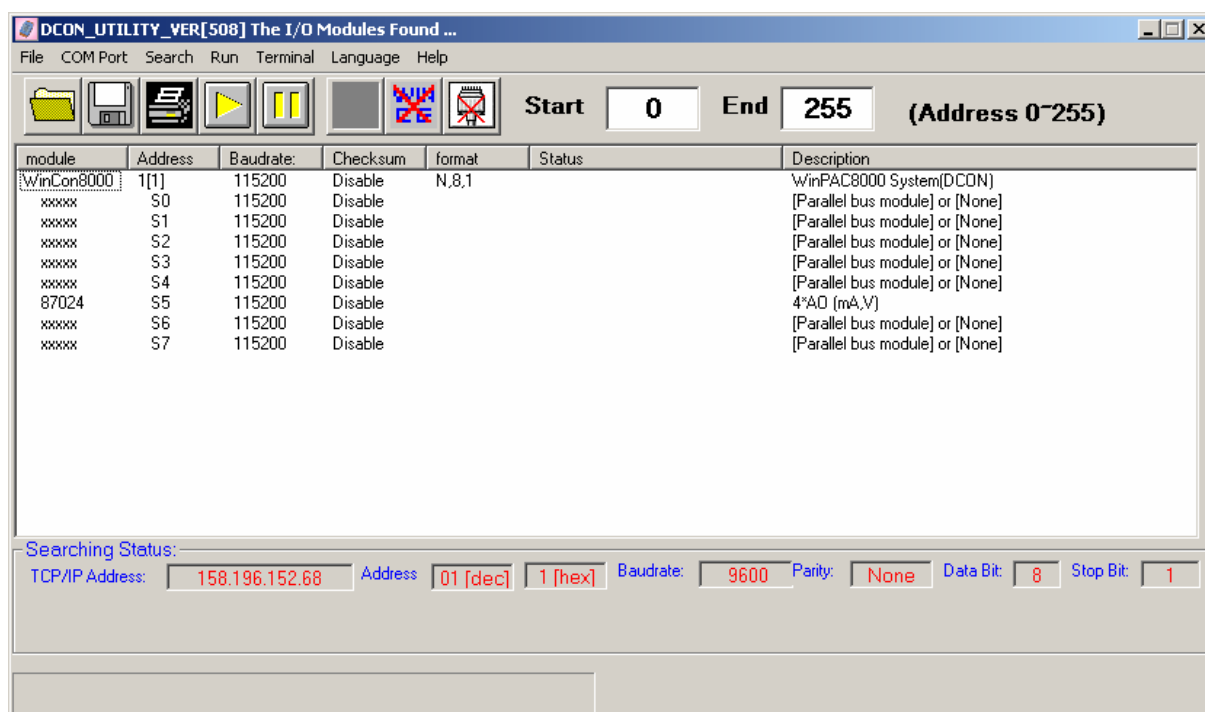
Před spuštěním aplikace na PC je nutné ověřit si, zda se ve WinConu v adresáři: *CompactFlash\ICPDAS\Tools\* nebo *Compact Flash\ICPDAS\Tools\...* nachází soubor *DCON\_CE\_V20x.exe*. V případě, že v daném umístění chybí, je nezbytné jej tam zkopírovat. Soubor slouží pro přenášení dat o nalezených modulech z Winconu do DCON Utility na PC, přičemž jeho spuštění se provádí automaticky bez zásahu operátora a to přes službu TelNet.

Při spuštění programu „DCOM\_Utility“ ani „I87k\_In\_Slot.exe“ nesmí běžet žádný řídicí program komunikující s I/O zásuvnými moduly - např. exekutiva řídicího systému REX. Toho je možné docílit dvěma způsoby: nastavením sériových zásuvných modulů již před instalací REXu na

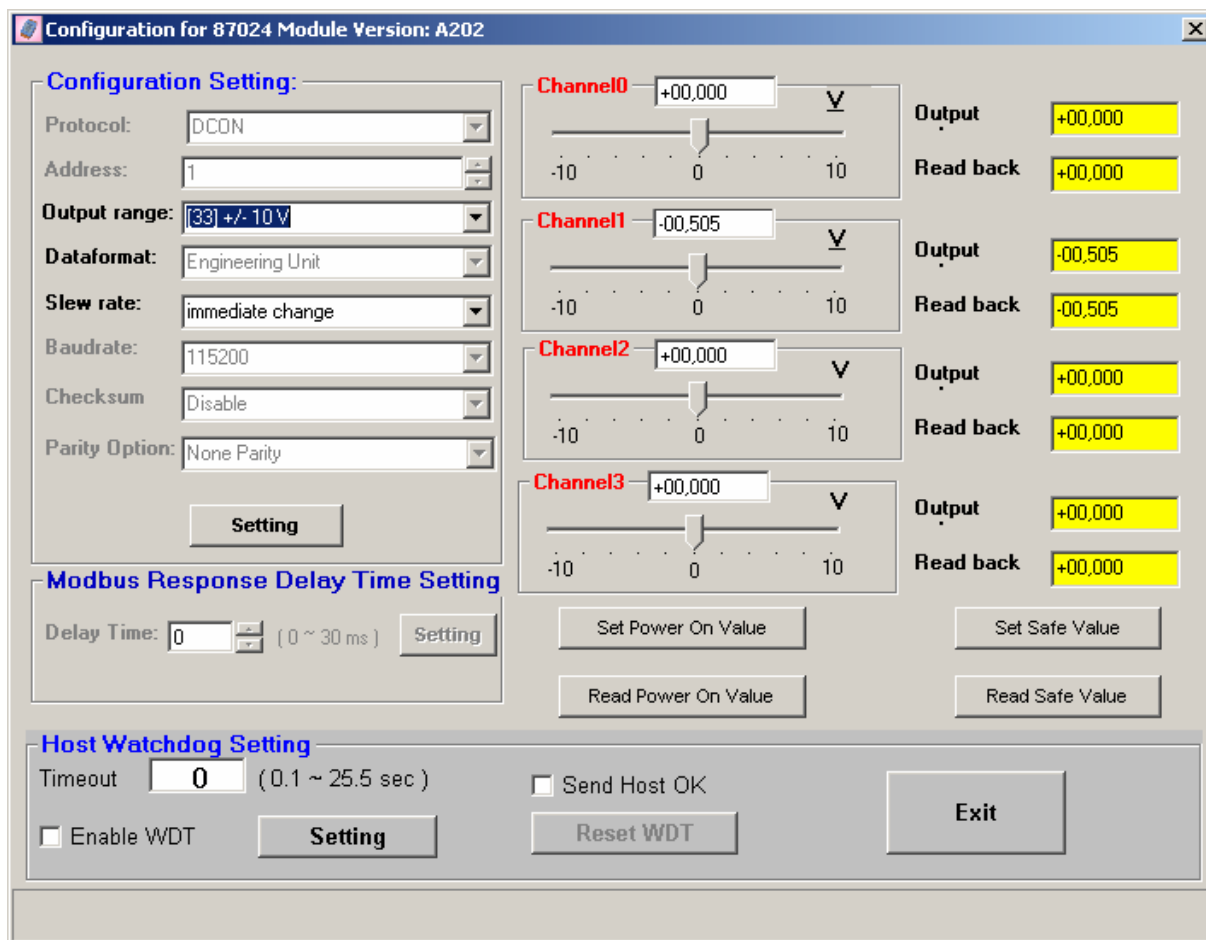
WinConu nebo zastavením běžící exekutivy z programu „RexView“ - menu „Target/Pause Execution“. Po nastavení modulů je možné exekutivu opět rozběhnout pomocí „Continue Execution“.

Jakmile jsou provedeny všechny předchozí kroky, lze na PC spustit aplikaci DCON Utility. Tlačítkem na liště se symbolem WINCE se zobrazí nabídka s možnostmi připojení. Zde se vybere záložka pro připojení k programu DCON\_CE\_V20x.exe pomocí sériového kabelu a ethernetu. Zadá se zde IP adresa WinConu a tlačítkem „Connect“ se vytvoří spojení. Pokud bylo vše správně nastaveno, program by měl sám začít vyhledávat moduly na zadaném portu sběrnice COM. Pro zobrazení modulů zasunutých přímo v základní jednotce WinConu je třeba v nastavení programu přepnout prohledávaný „COM port“ z „COM3“ na „COM1“.

V našem případě bylo potřeba změnit rozsah analogových výstupů modulu I-87024 se čtyřmi AO z původních 0 až 10 V na +-10V. Po připojení k WinConu byly prohledány a zobrazeny seřazené sériové zásuvné moduly zapojené ve WinConu (Obr. 6). Pro nastavení parametrů daného modulu se dvojklikem na daném modulu otevře nové okno s jednotlivými možnými nastaveními (Obr. 7).



Obr. 6: Výpis sériových zásuvných modulů ve WinConu na portu COM 1



Obr. 7: Nastavení sériového modulu I-87024

Pro inicializaci modulu I-87024 byla stažena poslední verze programu *DCOM Utility* – 5.0.6. Ta dle manuálu podporuje inicializaci modulů ve WinConu i samostatně přes ethernet při použití programu *dcon\_ce\_v210.exe* ve WinConu, nicméně tohoto se nepodařilo docílit. Program *DCOM Utility* vždy při prohledávání modulů přestal reagovat na příkazy. Nicméně při připojování a inicializaci přes ethernet spolu se sériovým kabelem se nakonec inicializace podařila. Přesto se i při tomto způsobu projevuje jedno úskalí – program *DCON\_CE\_V20x.exe* ve WinConu si při prvním spuštění „rezervuje“ porty COM1 a COM3 a při následném nebo opakovaném pokusu o znovu připojení se už spojení nevytvoří. Jediným řešením vzniklé situace je celý WinCon restartovat (a to buď přes „WinConu Utility“ nebo odpojením od napájecí sítě).

## **2.5 Další dostupné typy PAC výrobce ICPDAS**

Alternativou k WinConům jsou i další PAC od společnosti ICPAS. Protože se nejedná přímo o tentýž typ PAC jako je WinCon, nebyly tyto PAC uvedeny v kapitole popisující jednotlivé řady WinConu, na druhou stranu zastupují velmi podobnou kategorii zařízení PAC, ne-li jeho nástupce, proto budou v této práci zmíněny. Jedná se o modely ICPDAS WinPAC a XP-8000 a dále iPAC-8000. [3]

### **Kontrolér WinPAC**

Tento kontrolér WP-8x4x nazvaný WinPAC je vlastně navazující generací PAC kontroléru WinCon. Od něj se liší novějším operačním systémem Windows CE 5.0, rychlejším procesorem a jinými použitelnými moduly – v tomto případě s větší hloubkou tzv. High Profile. Konektivita rozhraní a portů je stejná jako u typu 8x4x. Stejně tak podpora hotových vývojových nástrojů je podobná jako u modelů WinCon, včetně řídicího systému REX.

### **Kontrolér XPAC 8000**

Zjednodušeně řečeno je XP-8000 vestavěný osobní počítač upravený pro účely řízení průmyslových procesů. Jde v současné době o nejnovější a nejvýkonnější PAC v nabídce společnosti ICPDAS. Je vybaven procesorem MD LX 800 CPU (500MHz) a operačním systémem Windows Embedded 2009 (což je nástupce MS Windows XP Embedded). Operační paměť RAM má 1GB, vestavěná paměť typu FLASH 4GB a je rozšiřitelné dalšími paměťovými moduly.

Co se týká hardwarové výbavy, je zde rozhraní VGA, USB, Ethernet, RS-232, RS-485. Pomocí rozšiřujících modulů dále umožňuje připojení přes sběrnici CAN a FrNet. XP-8000 existují ve verzích pro 0/3/7 zásuvných měřicích modulů přímo v základní jednotce, moduly jsou zde typu High Profile, tedy ty s větší hloubkou. Tyto moduly jsou v názvu označeny písmenem „W“ – například I-8017W je modul s 8AI. Díky novému operačnímu systému XPAC obsahují následující charakteristické funkce:

1. FTP Server
2. HTTP Server
3. IIS – Internetová informační služba
4. ASP .NET
5. SQL Server Express Edition 2005
6. .NET Framework 3.5
7. Vestavěný OPC Server (tzv. Quicker) pro přístup ke všem zásuvným modulům z libovolných aplikací.

Operační systém Windows Embedded Standard 2009 obsahuje „plnohodnotný desktopový“ .NET Framework 3.5. Díky tomu a sadě SDK, která je volně ke stažení ze stránek výrobce, je možné navrhovat řídicí algoritmy přímo ve Visual Studiu 2003/2005/2008. Systém zároveň umožňuje navrhnout ASP .NET webové stránky a zpřístupnit je na síti či internetu přes IIS. Všechny tyto aplikace mohou běžet již přímo v PAC, není nutný jakýkoliv další počítač typu PC.

Nyní uvažujme vzorovou aplikaci, kdy PAC bude řídit zvolený proces a jeho vizualizace bude formou ASP .NET stránek. V tomto případě tedy bude možné řízený proces monitorovat jak vzdáleně po síti přes webový prohlížeč, tak lokálně, například na dotykovém lcd-panelu připojenému přímo

k PAC. Tady lze návrh jak řídicího algoritmu tak webových stránek provádět ve Visual Studiu, pouze si navíc stáhneme SDK pro vývoj aplikací na XP-8000. SDK je volně ke stažení na webu ICPDAS. Pokud navíc použijeme Visual Studio ve verzi Express, není vlastně nutná žádná licence na jakýkoliv software (kromě licence OS na vývojovém PC, kde musí být OS MS Windows). Express edice Visual Studia je zdarma a na vzdáleném monitorovacím počítači může být nainstalován například OS Linux a webový prohlížeč Mozilla Firefox. V tomto případě by cena za celý systém byla pouze cena za PAC + měřicí moduly + práci programátora.

Podle dostupných informací z webu však zatím neexistují žádné „jiné“ vývojové prostředí, které by umožnilo návrh řídicích algoritmů pomocí buďto standardních PLC jazyků nebo jinými způsoby, které zrychlují návrh samotného algoritmu. To je způsobeno tím, že tento typ PAC je nový a zatím neexistují dané ovladače, například do řídicího systému REX, které by umožnily realizovat zápis a čtení I/O měřicích karet z PAC XP-8000.

Na PAC XP-8000 lze spustit OPC Server nazvaný „Quicker“. Jde o aplikaci, která umožňuje monitorovat jednotlivé I/O zásuvných modulů a zpřístupnit je lokálně i na síti v podobě jednotlivých položek (items) standardu OPC. K takovému serveru je následně možné připojit libovolný vizualizační software běžící buď přímo na PAC nebo na vzdáleném PC. Ačkoliv vizualizace není primárně určena pro výpočty typu zpětnovazebního řízení, jednodušší výpočty nebo jen monitoring v něm lze realizovat. Zde se však nabízí otázka, jak je to s kompatibilitou programů pro cílový operační systém Windows Embedded 2009. Tento OS byl vydán teprve nedávno a proto jsou jakékoliv informace zatím velmi vzácné. Jediný vizualizační software, který by měl běžet (dle vyjádření ICPDAS) je InduSoft.

Taktéž v současné době zatím není možné zjistit podrobnější informace o rychlosti odezvy aplikací PAC s OS Windows Embedded 2009, ani pro jaké minimální časy spouštění úlohy by se systém měl chovat deterministicky (real-time).

### **Kontrolér iPAC-8000 Compact PAC**

Kontrolér iPAC je podobný systému WinCon, obsahuje však jiný operační systém – je zde použito MiniOS7 (podobný systému DOS), pomalejší procesor i menší operační paměť. Konektivita zahrnuje pouze rozhraní ethernet a sériové RS-232 a RS-485, tedy systém neumožňuje připojit monitor s přímou vizualizací. Zásuvné měřicí moduly využívá stejného typu jako WinCon, paralelní a sériové, umístitelné buď přímo do základní jednotky nebo do rozšiřujících jednotek připojených přes RS-485. Systém je programovatelný několika způsoby. V prvním případě pomocí C kódu a SDK k tomuto PAC, dále pomocí software ISaGRAF a nakonec pomocí Matlab-Simulinku. Ke každému vývojovému nástroji je třeba jiný konkrétní typ tohoto PAC, jelikož ty obsahují runtime vždy pouze pro jeden typ vývojového nástroje. V případě varianty PAC pro Matlab je zároveň dodávána knihovna několika funkčních bloků pro Simulinku a přístup k jednotlivým I/O modulům případně systémovým událostem. Tyto bloky lze dále kombinovat s některými z bloků ze standardní knihovny Simulinku, přičemž ale vůbec nelze používat například bloky „Continuous“. Knihovna funkčních bloků i ovladače pro překladač schématu ze Simulinku do PAC jsou určeny pro verzi Matlabu 6.5, žádný novější podporován není, navíc vývoj software tohoto typu PAC již byl ukončen. Podstatná nevýhoda popisovaného řešení však je ta, že u řídicího schématu ze Simulinku přeloženého a nahraného do iPAC-8000 nelze zajistit komunikaci s žádným jiným zařízením například na síti nebo i jakýmkoliv OPC serverem, tudíž k takovému řídicímu systému není ani možné připojit žádnou vizualizační aplikaci.

### 3. Možnosti tvorby řídicích a vizualizačních aplikací

Na PAC zařízeních WinCon 8000 je možné vytvářet řídicí a vizualizační aplikace několika různými způsoby a programy. Některé zde budou popsány podrobněji, jiné budou pouze zmíněny. Z principu nahrávání řídicích algoritmů do WinConu je nutné si ujasnit, že je vybaven operačním systémem Microsoft Windows CE .NET. Tento „malý“ operační systém si vystačí s jednotkami až desítkami MB paměti a mimo procesorů x86 pracuje na různých RISC architekturách jako ARM (což je i případ WinConu) nebo MIPS. Jeho neopomenutelná vlastnost v oblasti řídicích systémů je především ta, že zaručuje deterministickou odezvu (hard real-time). Součástí operačního systému Windows CE je i .NET Framework, který tvoří vrstvu mezi aplikací .NET a operačním systémem. Využívá otevřené internetové standardy, jako XML, HTTP a SOAP. Jeho funkcí je, velmi jednoduše řečeno, zkompileovat aplikaci vytvořenou např. v jazycích VB, C++, C# do Intermediate Language (IL), jež je poté pomocí tzv. Just-In-Time kompilátoru (JITter) přeložena do strojového kódu spustitelného na dané platformě. Za předpokladu, že je k dispozici odpovídající JITter a dostatečně široká knihovna základních tříd, je aplikace v IL přenositelná mezi různými hardwarovými i softwarovými platformami. Kromě aplikace .NET Framework, určené pro Windows 2000/9x/NT, existuje i .NET Compact Framework, umožňující využívat služby .NET i na malých a mobilních zařízeních s Windows CE jako je WinCon. [6]

V případě tvorby řídicí aplikace ve Visual Studiu je možné pro vzdálené řízení a monitorování použít technologii iPush Embedded od společnosti ICE Technology. Podpora pro tuto technologii je již ve WinConu standardně integrována. Po spuštění iPush Embedded dojde k vytvoření tzv. iPush serveru na straně WinConu, jenž je schopen vzdáleným aplikacím odesílat data z připojených měřicích modulů a zároveň vyřizovat konkrétní požadavky na výstup.

Z hotových vývojových nástrojů pro tvorbu řídicích algoritmů lze využít relativně drahého software ISaGRAF. Ten je určen pro vývoj aplikací pro WinCon způsobem známým u většiny PLC, tj. jedním z pěti standardních jazyků definovaných normou IEC61131-3 (LD, ST, FBD, SFC, IL). Další možností je řídicí systém REX firmy REX Controls, který umožňuje grafický návrh řídicího algoritmu přímo v prostředí Matlab-Simulink, odkud je možné i celý algoritmus simulovat bez nutnosti zkoušení přímo na hardwaru WinConu a připojené technologii.

Pro vizualizaci procesů lze využít nástroje InduSoft WebStudio, který kromě tvorby webových aplikací, umožňuje i návrh a tvorbu vizualizačních aplikací pro různé operační systémy (např. Windows CE nebo XP). Pro vizualizaci na PC je možno využít jakýkoliv známý vizualizační software jako třeba Promotic. K napojení na oba dříve jmenované řídicí systémy pak slouží standard OPC včetně serveru, který je vždy nedílnou součástí REXu.

#### 3.1 SDK pro WinCon na platformě .NET Framework

Platforma Microsoft Windows CE 4.1 s .NET Compact Frameworkem 1.0 umožňuje návrh, implementaci a generování kódu programů přímo ve vývojových prostředích pro tyto „Windows“ jako jsou Microsoft Visual Studio a Embedded Visual Studio. V případě Visual Studia je nutné využít verze 2003, jelikož všechny novější edice již nepodporují starší .NET Compact Framework ve verzi 1.0. Ve Visual Studiu je možné vytvářet jak řídicí algoritmy programu, tak vytvořit vizualizační okno, neboli panel, přes které bude možné tuto aplikaci ovládat. Výrobce ICPDAS zpřístupnil volně

ke stažení SDK (Software Development Kit – sada nástrojů pro vývojáře) a vzorové příklady kódů pro tyto již dříve jmenovaná vývojová prostředí. Součástí SDK jsou pak například .dll knihovny, které se připojují k projektu ve Visual Studiu a slouží pro přístup k hardwaru WinConu jako jsou vstupně/výstupní zásuvné měřicí moduly, systémové informace a podobně. U tohoto způsobu návrhu řídicích algoritmů je však zásadní nevýhoda jak samotná složitost návrhu tak především časová náročnost. Jakýkoliv, i poměrně jednoduchý program navržený pro řízení určité strojní technologie zaměstná programátora neporovnatelně delším časem, než kdyby byl použit již hotový vývojový software typu IsaGraf nebo REX. Každopádně zpřístupnění této SDK umožnilo těmto i jiným firmám navrhnout vývojový řídicí software určený speciálně pro WinCony nebo alespoň zahrnout podporu WinConu do jejich vývojových produktů. [7]

Platforma .NET Framework ve WinConu standardně podporuje webový http server takže je možné tímto způsobem monitorovat či sledovat třeba aktuální stav určitého procesu. Bohužel lze takto zobrazovat pouze pasivní webové stránky html, systém totiž plně nepodporuje stránky ASP (Active Server Pages) a dále ani Internetovou informační službu IIS. Pro zpřístupnění některých souborů na síti lze využít FTP server který je taktéž standardně integrován. Veškerá naměřená data pak lze ukládat do databáze typu SQL CE do WinConu, ke které ale není možný přístup programů z jiných počítačů na ethernetové síti. Pomocí funkce TELNET se lze vzdáleně připojit k WinConu a spouštět libovolné programy. [9]

### **3.2 iPush Embedded**

iPush Embedded je průmyslová verze ICE iPush Communication Serveru vyvinutého společností ICE Technology Corporation pro síťovou komunikaci dvou a více aplikací typu server – klient, přičemž server zde běží na platformě CE.NET ve WinConu. [3]

iPush Embedded představuje způsob přenosu dat po platformě TCP/IP, který umožňuje přijímat i odesílat data vzdáleným aplikacím v reálném čase. Všeobecně je možné pomocí této technologie vyměňovat jakákoliv data nebo proměnné typu string, integer, real a boolean. Ve WinConu pak slouží především pro vzdálený přímý přístup k I/O zásuvným modulům řady I-8000, I-87K a I-7000. Data sesbíraná z těchto I/O modulů mohou být aktivně odesílána ke zpracování vzdáleným aplikacím, ty jsou zase schopny vracet instrukce pro přímé ovládání I/O modulů. iPush Embedded může také zajistit výměnu dat mezi skupinou navzájem komunikujících zařízení WinCon-8000 apod.. Mimoto lze pomocí technologie iPush Embedded vzdáleně provádět základní nastavení (vzdálenou administraci) zařízení WinCon. Systém přenosu dat pak sestává z ICE iPush Communication Server Embedded na straně WinConu-8000 a klientské aplikace na PC či mobilním zařízení typu PDA a podobně. Standardně je iPush Server naprogramován od výrobce ICPDAS tak, že odesílá klientským aplikacím hodnoty na vstupech jednotlivých vstupně-výstupních zásuvných modulů a zároveň od klientských aplikací přijímá požadavky na zápis hodnoty na dané výstupu zásuvných modulů a provádí je, případně umožňuje vzdálenou administraci nebo pouze identifikaci jednotlivých zásuvných modulů.

Klientská aplikace využívá ActiveX komponentu iPush, kterou je potřeba před jejím prvním použitím ve Windows XP registrovat. Toto lze provést například v příkazovém řádku pomocí příkazu „regsvr32 iceHMsg.ocx“ a „regsvr32 iPushX.ocx“. Při programování klientské aplikace ve Visual Studiu se pak využívá ActiveX komponenta iPush, přes jejíž rozhraní samotná komunikace se serverem probíhá. Tuto komponentu je nutné známým způsobem připojit do vytvářeného projektu



ve Visual Studiu. Poté lze funkcemi jejího rozhraní komunikovat s iPush serverem. Program lze tedy vytvářet v libovolném programovacím jazyce, samozřejmě podporovaným danou verzí Microsoft Visual Studia. Výrobce ICPDAS dodává několik vzorových programových kódů a aplikací psaných v jazyce Visual Basic nebo C++ pro vzdálené řízení I/O modulů, vzdálenou administraci nebo pouze identifikaci jednotlivých zásuvných modulů.

Nevýhoda tohoto řešení je však zřejmá. Standardně lze komunikovat mezi zařízeními pouze hodnotami na vstupně-výstupních zásuvných modulech, vyměňovat například hodnoty proměnných v paměti takto nelze. K tomu by bylo nutné upravit část programového kódu iPush serveru, což podle podkladů ICPDAS lze, nicméně jde o poměrně složitý a především časově náročný proces. [10]

### **3.3 Matlab – Simulink**

Simulink je program pro simulaci a modelování dynamických systémů, který využívá algoritmy MATLABu pro numerické řešení nelineárních diferenciálních rovnic. Poskytuje uživateli možnost rychle a snadno vytvářet modely dynamických soustav ve formě blokových schémat a rovnic.

Pomocí Simulinku a jeho grafického editoru lze vytvářet modely lineárních, nelineárních, v čase diskretních nebo spojitých systémů pouhým přesouváním funkčních bloků myší. Simulink nově umožňuje spouštět určité části simulačního schéma na základě výsledku logické podmínky. Otevřená architektura dovoluje uživateli vytvářet si vlastní funkční bloky a rozšiřovat již tak bohatou knihovnu Simulinku. Hierarchická struktura modelů umožňuje koncipovat i velmi složité systémy do přehledné soustavy subsystémů prakticky bez omezení počtu bloků. Simulink, stejně jako MATLAB, dovoluje připojovat funkce napsané uživateli v jazyce C. Mezi neocenitelné vlastnosti Simulinku patří nezávislost uživatelského rozhraní na počítačové platformě. Přenositelnost modelů a schémat mezi různými typy počítačů umožňuje vytvářet rozsáhlé modely, které vyžadují spolupráci většího kolektivu řešitelů na různých úrovních.

Otevřená architektura Simulinku vedla ke vzniku knihoven bloků, nazývaných *blocksets*, které rozšiřují základní knihovnu bloků Simulinku a umožňují použití programu v příslušných vědních a technických oborech. Knihovny je možné rozšiřovat i o vlastní bloky, vytvořené uživatelem. V případě instalace řídicího systému REX, který bude popsán podrobněji v další kapitole, se provede právě přidání nových bloků z knihovny REXLib do knihovny bloků v Simulinku. Tyto bloky je možné využívat v simulacích a na tomto systému je v podstatě řídicí systém REX založen. [12]

### 3.4 Řídicí systém REX

REX je multiplatformový řídicí systém kompatibilní s globálně rozšířeným simulačním systémem Matlab-Simulink. V současné době je REX implementován na platformě Windows (nejlépe 2000/XP/Vista), Windows CE .NET a operačním systémem reálného času Phar Lap ETS.

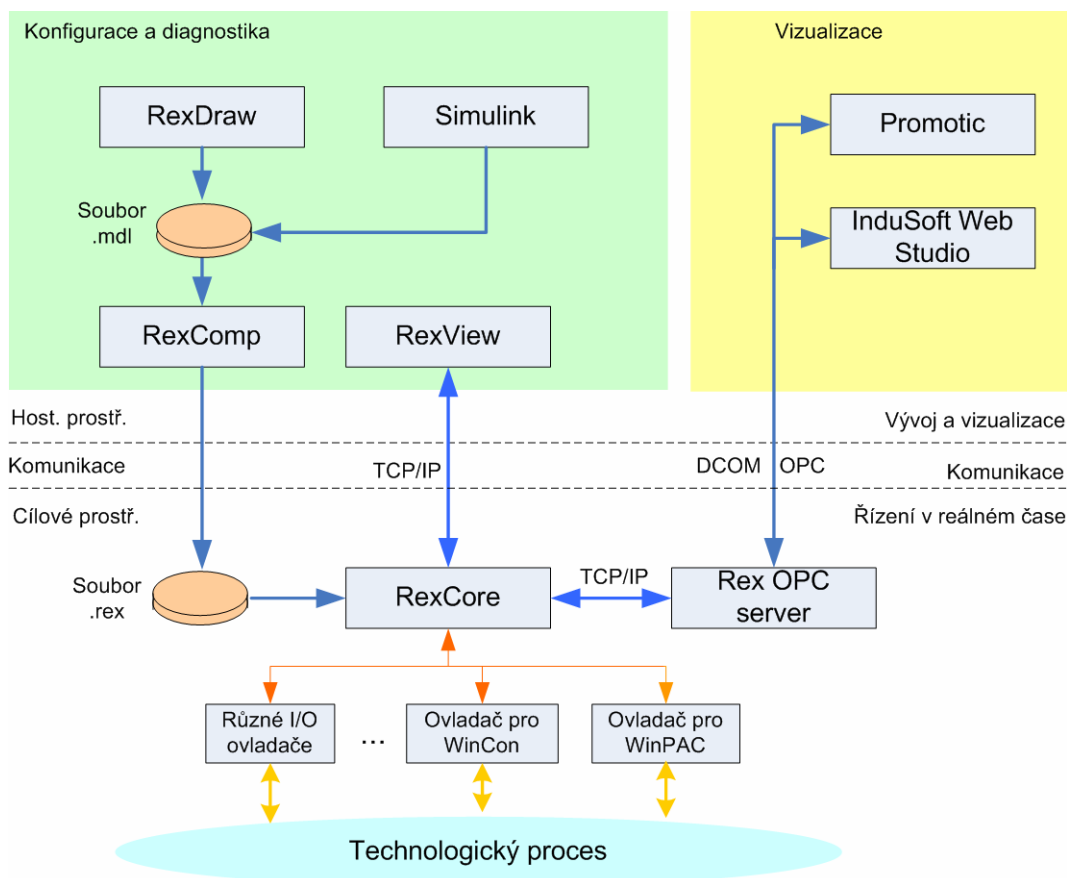
Kompatibilita s rozšířeným simulačním systémem Matlab-Simulink je hlavní myšlenkou řídicího systému REX. Tím uživatel získává všechny jeho výhody pro návrh vyspělých algoritmů řízení. Řídicí algoritmy lze navrhovat graficky přímo v prostředí Matlab-Simulink a zde je hned simulovat. Pro plnohodnotné využití systému REX však licence systému Matlab-Simulink není nutná, protože tytéž algoritmy lze vytvářet i v návrhovém programu RexDraw, jenž je vždy součástí systému REX. Další důležitou vlastností je implementace standardního rozhraní OPC umožňující napojení na všechny významné systémy pro vizualizaci a operátorské ovládání řízených procesů. [5]

#### 3.4.1 Současné podporované platformy ŘS REX:

- Počítače PC a IPC (průmyslová PC) s OS Windows 95/98/ME/NT4/2000/XP/Vista, Phar Lap ETS a VenturCom RTX.
- WinPLC s moduly řady DL205 s OS Windows CE
- TECOMAT TC700 s OS Windows CE
- ICPDAS WinCon 8x3x s OS Windows CE .NET
- ICPDAS WinPLC s OS Windows CE 5.0

#### 3.4.2 Struktura řídicího systému REX

Řídicí schémata lze navrhovat buďto v Matlab-Simulinku, nebo v programu RexDraw, který je součástí instalace řídicího systému REX. Jedním z těchto způsobů je vytvoření řídicího schéma v podobě souboru .mdl, které se následně přeloží překladačem RexComp do souboru .rex. Tento soubor je možné buď přes diagnostický program RexView nebo přímo přes Simulink nahrát do jádra systému REX, což je RexCore. RexCore již běží na cílovém zařízení přičemž vykonává jednotlivé činnosti související s během řídicích schémat. Vizualizační nástroje pak komunikují s jádrem RexCore přes OPC server. Historické trendy i archivy lze vyčítat přes rozhraní Automation například do Excelu. Základní struktura řídicího systému REX je zřejmá z Obr. 8. [13]



Obr. 8: Struktura řídicího systému REX

### 3.4.3 REX + WinCon 8x31 (Windows CE)

REX s průmyslovým kompaktním regulátorem WinCon řady 8x31 tvoří řešení vhodné pro rozsáhlý sběr dat a implementaci náročných algoritmů řízení. Ve srovnání s kompletním HW+SW řešením jiných výrobců PC + PLC je výrazně levnější. U jednoduchých úloh je možné dosáhnout periody vzorkování až 2 ms.

Instalace řídicího systému REX na platformu WinCon je možná několika způsoby. Asi nejjednodušší je nahrát na paměťovou kartu Compact Flash ve WinConu instalační soubor REXu, který je volně ke stažení na webu výrobce, bez licence je však omezena doba jeho běhu na dvě hodiny, poté je nutno systém REX opětovně spustit. Samotné nahrání na paměťovou kartu lze provést pomocí USB Flash disku nebo přes FTP Server ve WinConu. Spuštěním daného instalačního souboru se řídicí systém sám REX nainstaluje, zapíše do registrů potřebná data a nastaví se jako aplikace automaticky spouštěná po startu operačního systému na WinConu. Nyní je nutné pomocí aplikace *WinCon Utility* provést uložení registrů a restart celého zařízení. Po následném naběhnutí operačního systému již řídicí systém REX běží na pozadí. V případě, že není zadán licenční klíč, je vždy po startu nezbytné potvrdit hlášku o běhu REXu v demonstračním dvouhodinovém režimu. [13]

### 3.4.4 Řídicí systém REX a vstupně-výstupní ovladače

Systém REX je otevřeným průmyslovým řídicím systémem kompatibilním se simulačním systémem Matlab-Simulink. Tato kompatibilita je založena na využití formátu souborů s modely simulovaných systémů .mdl i pro konfiguraci systému REX a dále na knihovně funkčních bloků RexLib, která existuje jak pro Matlab-Simulink tak pro každou cílovou platformu systému REX. [14]

Jedním z rozšíření systému REX oproti systému Matlab-Simulink je vstupně-výstupní subsystém pro připojení řídicích algoritmů na vstupy a výstupy reálných procesů. Toto připojení umožňují realizovat právě vstupně-výstupní ovladače systému REX. Řídicí systém REX je obecně implementován pro dvě prostředí:

#### 1. Vývojové prostředí (Host).

Slouží pro konfiguraci (program RexDraw), překlad (program RexComp), diagnostiku (program RexView) a případnou vizualizaci (OPC server RexOPCsv a vizualizační programové vybavení) konkrétní aplikace řídicího systému REX. Vývojovým prostředím jsou počítače PC s operačními systémy Windows NT4/2000/XP/Vista).

#### 2. Cílové prostředí (Target).

Umožňuje provozovat vytvořenou aplikaci řídicího systému REX na cílovém zařízení (program RexCore). Cílovým prostředím být mohou např. Embedded PC, PLC apod. s operačními systémy Windows 95/98/ME/NT4/2000/XP, Windows CE .NET apod.

Pro operační systémy Windows může být vývojový počítač současně i cílovým, což nijak neodporuje výše uvedenému rozdělení. Funkce základních programů řídicího systému REX je rozšiřitelná pomocí tzv. modulů. Každý vstupně-výstupní ovladač systému REX musí být součástí nějakého modulu, jeden modul však může obsahovat ovladačů několik (moduly jsou v operačních systémech s rozhraním Win32 implementovány pomocí tzv. DLL knihoven (Dynamic-Link Library)).

Pro ovládání vstupů a výstupů jednotlivých modulů zařízení série WinCon 8x31 je v systému REX určen ovladač "WcnDrv". Pro správnou funkci ovladače "WcnDrv" v řídicím systému REX je zapotřebí provést následující kroky:

1. Zařadit ovladač do projektu aplikace řídicího systému REX
2. Zkonfigurovat ovladač z programu RexDraw nebo Matlab-Simulink standardním způsobem.

Vývojový modul ovladače "WcnDrv" se instaluje jako součást instalace řídicího systému REX pro platformu Windows. Po úspěšné instalaci se do instalačního adresáře zkopírují soubory:

WcnDrv\_H.dll (Konfigurační část ovladače WcnDrv)










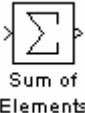




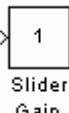
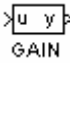
DOC\WcnDrv\_MULTI\_CZ.pdf (uživatelská příručka)


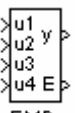
















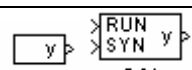

WcnDrv\_T.dll (Cílová část ovladače WcnDrv spouštěná exekutivou reálného času)

### 3.4.5 Knihovna funkčních bloků RexLib:

Řídicí systém REX obsahuje rozsáhlou knihovnu funkčních bloků RexLib, v současné době tvořenou více než 100 bloky. Bloky jsou vytvořeny jak přímo pro Matlab-Simulink, tak pro každou cílovou platformu. V případně simulačních schématach v Simulinku je možné bloky z RexLib libovolně kombinovat s jakýmkoliv jinými bloky, pouze v nastavení parametrů simulace je třeba použít typ „Fixed Step“ a „Discrete time simulation“. Pokud však bude schéma nahráno do řídicího systému REX, je nutné využívat pouze bloky z RexLib, jakékoliv jiné bloky budou při překladu schématu pomocí překladače RexComp vypuštěny. Tímto se dostáváme k podstatnému omezení REXu a to, že řídicí algoritmus lze navrhovat pouze s bloky z knihovny RexLib. Na druhou stranu ke většině bloků ze standardní knihovny Simulinku lze nalézt i jejich odpovídající náhrady z knihovny RexLib. Proto byl v rámci diplomové práce vytvořen přehled náhrad standardních bloků Simulinku bloky z knihovny RexLib. Celý přehled je součástí přílohy diplomové práce (viz Příloha I), příklad vybraných alternativ je v následujících tabulkách. V každé tabulce je vždy schéma jednotlivých bloků ze Simulinku a jejich možná náhrada z knihovny REXu, případně poznámka k využití daných bloků. [14]






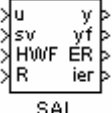

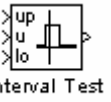
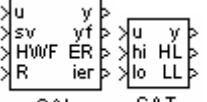





Math Operations:

Simulink	REX	Pozn.
 Sum	 ADD  SUB	
 Add	 ADD  EAS	Blok „EAS“ pro více vstupních signálů
 Subtract	 SUB  EAS	Blok „EAS“ pro více vstupních signálů
 Sum of Elements	Blok nemá v REXu alternativu.	V REXu na jednom „drátě“ pouze jedna proměnná (nevektorová, nematicová)
 Bias	 ADD	Offset lze nahradit blokem „ADD“
 Gain	 GAIN	
 Slider Gain	 GAIN	totéž jako „Gain“ pouze s možností úpravy zesílení za běhu simulace pomocí „slideru“ ve vlastním okně

 Product	 EMD	násobení a dělení vstupních signálů
 Divide	 DIV	
 Product of Elements	Blok nemá v REXu alternativu.	
 Dot Product	Blok nemá v REXu alternativu.	na vstupu předpokládá vektory, nelze takto vytvořit alternativu
 Sign	Blok nemá v REXu alternativu.	
 Abs	 ABS_	
 Unary Minus	 GAIN	není přímá alternativa, lze obejít přes „Gain“ s parametrem „-1“
 Math Function	 FNX	
 Polynomial	 POL	
 Trigonometric Function	 FNX	Alternativa pouze přes blok „FNX“ s výběrem dané funkce sinus
 Sine Wave Function	 SG SGI	
 Algebraic Constraint	Blok nemá v REXu alternativu.	

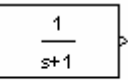
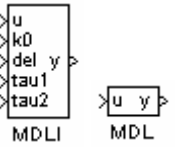
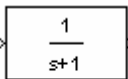
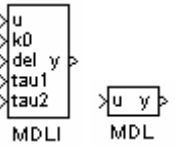
**Tabulka 6: Přehled možných náhrad standardních bloků Simulinku z knihovny Math Operations bloky z knihovny RexLib**

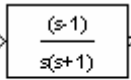
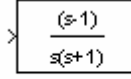
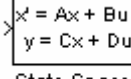

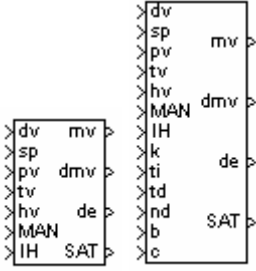

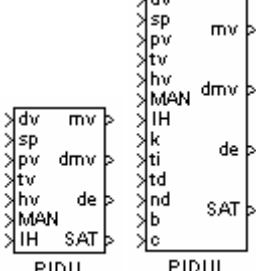
## Logic Operations:

Simulink	REX	Pozn.
 Logical Operator	 AND_	
 Relational Operator	 CMP	
 Interval Test	 SAI	Blok „SAI“ lze použít mimo jiné i pro zjištění, zda je hodnota proměnné v daném rozsahu.
	 SAT	
 Interval Test Dynamic	 SAI SAT	
 Combinatorial Logic	Blok nemá v REXu alternativu.	
 Compare To Zero	 CMP	
 Compare To Constant	 CMP	

**Tabulka 7: Přehled možných náhrad standardních bloků Simulinku z knihovny Logic Operations bloky z knihovny RexLib**

## Simulink Extra: Additional Linear

Simulink	REX	Pozn.
 Transfer Fcn (with initial states)	 MDLI MDL	
 Transfer Fcn (with initial outputs)	 MDLI MDL	

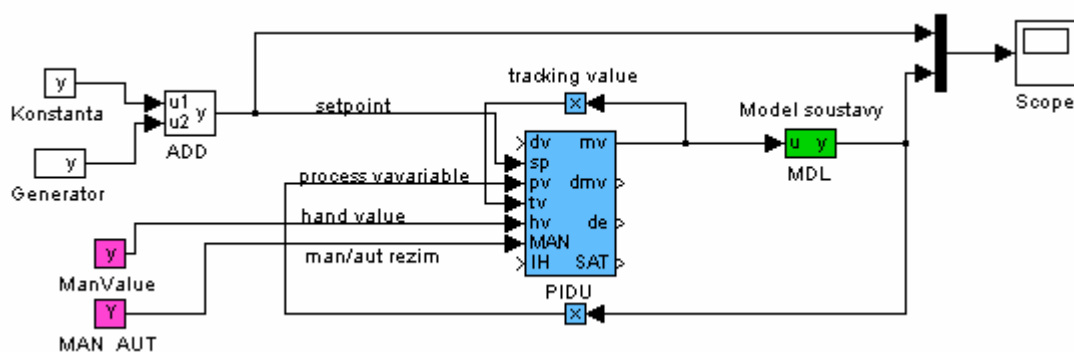
 <p>Zero-Pole (with initial states)</p>	Blok nemá v REXu alternativu.	
 <p>Zero-Pole (with initial outputs)</p>	Blok nemá v REXu alternativu.	
 <p>State-Space (with initial outputs)</p>	Blok nemá v REXu alternativu.	
 <p>PID Controller</p>	 <p>PIDU      PIDUI</p>	V REXLib existují PID regulátory s manuální režimem, potlačením wind-up efektu, autotunerem, saturací výstupu apod.
 <p>PID Controller (with Approximate Derivative)</p>	 <p>PIDU      PIDUI</p>	

**Tabulka 8: Přehled možných náhrad standardních bloků Simulinku z knihovny Simulink-Extra: Additional Linear bloky z knihovny RexLib**

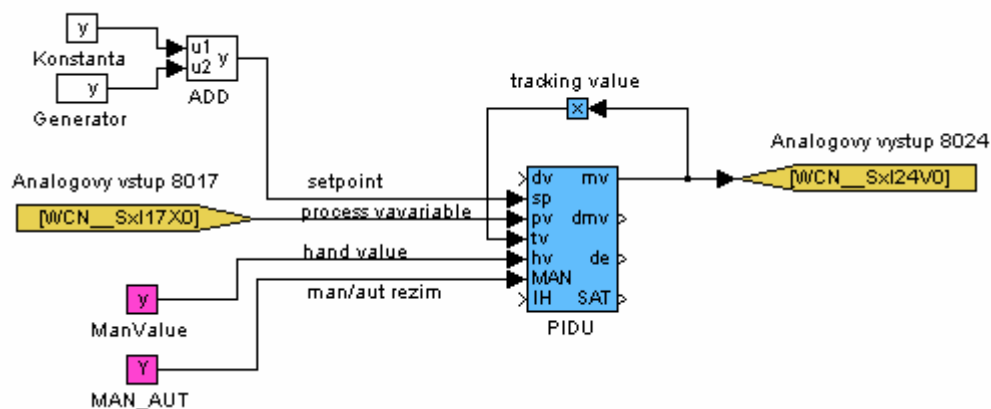


### 3.4.6 Příklad konfigurace

Níže uvedený Obr. 9 ilustruje konfiguraci jednoduché regulační smyčky s PID regulátorem typu PIDU, ke kterému je připojena simulační soustava. Zobrazené regulační schéma lze ladit a testovat v Simulinku. Jakmile je nalaďení regulátorů a dalších částí regulační smyčky hotovo, je možné přikročit k dalšímu kroku, tedy zařazení skutečné soustavy do projektu. Toto ilustruje další schéma (viz Obr. 10) s připojenou skutečnou soustavou na jednotlivé I/O kanály zásuvných modulů WinConu.



Obr. 9: Regulační schéma s modelem soustavy pro simulaci procesu řízení



Obr. 10: Regulační schéma se skutečnou soustavou připojenou na I/O WinConu

### **3.4.7 OPC Server v instalaci REXu**

Součástí instalace REXu 1.41 pro Windows XP/CE/Vista je i OPC Server verze 2.0 skládající se z následujících součástí:

#### **REX OPC Server**

Zpřístupňuje hodnoty všech parametrů bloků ve schématech běžícího programu v REX Runtime. Je možné jak měnit vnitřní parametry bloků tak ovlivňovat vstupy daného bloku a sledovat hodnoty z něj vystupující. Tento OPC Server je dodáván ve formě objektu .exe souboru a nelze jej provozovat na operačním systému Windows CE.

#### **REX inproc OPC Server**

Tento OPC server typu in-process je dodáván ve formě aktivní .dll knihovny. Jeho využití je podobné jako v případě REX OPC Serveru. Je však možné jej provozovat jak na PC s operačními systémy Windows XP/Vista, tak na Windows CE a tedy i na WinConu. V takovém případě však OPC Server běží na kontextu lokálního klienta, tedy není možné jej spouštět vzdáleně z klientů na okolních PC nebo zařízení na stejné ethernetové síti. Tento typ OPC serveru je jako jediný ze zde uvedených totožný svým chováním jak ve verzi na PC tak ve verzi pro WinCon.

#### **RDC OPC Server**

Tento server funguje jako protipól bloku RDC ve schématu programu řídicího systému REX na daném počítači. V tom případě umožňuje přístup ke všem vstupům i výstupům komunikačního bloku RDC ve schématu programu.

#### **Testing OPC Server**

Obsahuje implicitně přednastavené proměnné pro testování připojení k OPC Serveru. Jednak statické „Value“ – jejichž hodnoty se v čase samovolně nemění. Dále proměnné „Oscilator“ jejichž hodnoty systém REX průběžně přenastavuje a je tedy vhodné je v klientovi pouze vyčítat.

Samotný OPC Server je automaticky spouštěn při požadavku jakéhokoliv OPC klienta podporujícího rozhraní OPCv2.0 a novější na soupis jeho dostupných veličin, které může přenášet (starší OPC servery toto neumožňují). V případě prostředí Promotic je možné tímto způsobem procházet jednotlivé veličiny nabízené OPC serverem.

Řídicí systém REX dále umožňuje připojení jiného OPC serveru do schématu řídicího programu. K tomu potřeba připojení ovladače OPC k exekutivě EXEC a nadefinování odebíraných a nastavovaných prvků (items) z tohoto OPC serveru. Přístup k těmto proměnným je pak v programu přes přesně definovanou konvenci parametrů vstupně výstupních vlajek. Přesný popis je v příručce „Ovladač OPCDrv systému REX pro Windows - Uživatelská příručka“.

V případě provozování řídicího systému REX na WinConu lze využívat DLL verzi OPC serveru nazvanou RexOPCin.dll. To však předpokládá, že OPC server běží na kontextu klienta, tj. v rámci klientského procesu. Tímto způsobem lze připojit například vizualizační software Indusoft Web Studio k řídicímu schématu. OPC server musíme na WinConu nejprve zaregistrovat to lze provést ručně příkazem „\Windows\RegSvrCE.exe \CompactFlash\REX\RexOPCin.dll“ kde RegSvrCE je systémový program z adresáře \Windows na WinConu nebo jednodušeji spuštěním souboru

„OPC\_reg.bat“. Tím se vytvoří příslušné klíče v registru, které je nejlépe hned uložit v programu WinCon Utility.

Druhou možností je provozovat REX na Winconu a OPC server na PC, na kterém poběží vizualizační aplikace – například Promotic nebo Indusoft Web Studio. Pak se však musí nastavit cílové zařízení, ke kterému se bude RexOPCsvg.exe připojovat, jinak RexOPCsvg hledá systém REX na lokálním počítači. Nastavení cílového zařízení se provádí např. příkazem (z příkazové řádky):

*RexOPCsvg -T=wincon*

přičemž do souboru "hosts" v adresáři windows\system32\drivers\etc se přidá IP adresu WinConu. OPC server umožňuje připojení se najednou i k několika systémům REX pokud jsou jednotlivá cílová zařízení oddělena středníky. Například výraz v příkazové řádce „RexOPCsvg -T=;wincon“ (tj. mezi znakem = a středníkem je prázdný řetězec, bude OPC server hledat systém REX na lokálním počítači a dále na zařízení WinCon. Potom příklad kódu souboru „hosts“ může mít tvar:

```
# Copyright (c) 1993-1999 Microsoft Corp.
# Toto je ukázka souboru HOSTS používaného službou Microsoft TCP/IP for Windows.
# ...
127.0.0.1      localhost
10.0.0.4      ucebnah310
10.0.0.6      wincon
```

Nastavení pomocí příkazu -T lze najít v registry (spouští se pomocí „regedit“) v klíči:

HKEY\_LOCAL\_MACHINE\SOFTWARE\REX Controls\REX\RexOPCsvg\TargetName

Standardně OPC servery (jak RexOPCsvg.exe, tak i RexOPCin.dll) komunikují přes TCP/IP, a pokud není nastaveno jinak, komunikují s lokálním počítačem, kde jsou spuštěny. V případě spuštění OPC serveru na lokálním WinConu, je možné navíc komunikovat rychleji, přes sdílenou paměť. Cílové stanice, se kterými komunikuje OPC server na daném počítači (i na WinConu) jsou uloženy v registry v klíči „HKEY\_LOCAL\_MACHINE\SOFTWARE\REX Controls\REX\RexOPCsvg“ v proměnné TargetName. Pokud je tato proměnná nenastavena, bude OPC server komunikovat s lokálním počítačem pomocí TCP/IP. Má-li OPC Server komunikovat přes sdílenou paměť je nutné nastavit do proměnné TargetName znak @, což je určitá konvence pro REXovské OPC servery.

Některé zde popsané způsoby připojování se k OPC Serveru REXu i jeho nastavování ve WinConu nejsou uvedeny v uživatelské příručce a byly zjištěny emailovou korespondencí se společností REX Controls.

### 3.4.8 Trendy, archivace a přenos procesních dat REXu do Matlabu :

Řídicí systém REX umožňuje záznam hodnot technologických proměnných do trendů. Takto mohou být ukládány buď do operační paměti daného zařízení, nebo i archivovány na paměťovou kartu ve WinConu či pevný disk v případě PC. K tomuto slouží blok „TRND“. Jeden blok umožňuje záznam až čtyř průběhů, přičemž počet těchto bloků použitých ve schématu není nijak omezen. Uložené průběhy z trendů lze následně vyčítat pomocí REXView či REXTrend nebo přes rozhraní Automation přenášet například do Excelu.

V případě pomalejších nekritických procesů, lze přenášet data přímo do Simulinku pomocí jednoho či několika bloků RDC (Remote Data Communication) z podknižovny Spec. Komunikace funguje tak, že vždy mezi sebou komunikuje dvojice bloků RDC. Jeden je nakreslen do schématu v REXu, druhý ve schématu v Simulinku. Bloky mezi sebou komunikují pomocí protokolu UDP/IP a vyměňují si data tak, že vstupy jednoho bloku RDC jsou přenášeny na výstupy druhého bloku RDC a obráceně, tj. "křížem". Na stranu Simulinku pak lze doplnit bloky Scope nebo displeje apod. nebo data posílat pomocí ToWorkspace do pracovního prostoru Matlabu. [14]

### **3.5 Indusoft Web Studio**

InduSoft Web Studio (IWS) je vizualizační software využívající klíčové vlastnosti operačních systémů Microsoft® Windows® NT/2000/XP a Windows® CE, které umožňují vytvářet WEB orientované SCADA a HMI aplikace pro řízení technologických procesů bez použití dalších nástrojů. InduSoft Web Studio dále umožňuje vytvářet lokální i webové aplikace v jednom vývojovém prostředí. S pomocí libovolného internetového prohlížeče lze vzdáleně spravovat IWS aplikace, monitorovat technologické procesy, přehrávat software a online přistupovat v reálném čase k datům jako jsou dynamické obrazovky, trendy, alarmy, reporty apod. Pomocí tohoto nástroje lze vytvářet vizualizační programy přímo pro platformu WinCon.

InduSoft Web Studio podporuje všechny základní průmyslové standardy jako jsou platformy Microsoft .NET, komunikace OPC, DDE, ODBC, formáty XML, SOAP a komponenty ActiveX. [3]

### **3.6 Virtual CE 4.2 Pro – vzdálená plocha**

Virtual CE Pro (dále jen VCEP) slouží pro vzdálenou správu a ovládání WinConu-8000. Jedná se o aplikaci, která především umožňuje přenášet snímky z obrazovky ve WinConu do PC. VCEP se skládá ze dvou částí: serveru, který běží ve WinConu a klienta běžícího na PC. Poté co je vytvořeno spojení mezi klientem a serverem, klient periodicky odesílá žádosti o znovu vykreslení obrazovky a zároveň odesílá informace o kliknutí myši či zmáčknutí tlačítka serveru k simulaci ve WinConu. Při přenosu jednotlivých snímků je nejdříve každý obrazový snímek porovnán s předcházejícím a teprve následně zkomprimován a odeslán klientovi, to pro maximální snížení toku přenesených dat. Rychlost překreslování obrazovky lze nastavit v rozsahu 1 až 18 snímků za sekundu. Limitující je samozřejmě rychlost zařízení v závislosti na vytíženosti systému a typu a podmínkách připojení. Podporovány jsou rozhraní sériový port, USB, ethernet a komunikace přes ActiveSync. Aplikace navíc umožňuje zachytávání snímků z WinConu přímo do formátů BMP, GIF nebo JPG a ukládat do AVI videosekvencí a přenos souborů do WinConu přes integrovaný souborový manažer. Ten zobrazuje systémovou strukturu ve WinConu a lze jej využít pro přenos souborů z lokálních disků PC nebo sítě do WinConu anebo obráceně do PC. [3]

Z praktického hlediska není tato aplikace příliš použitelná pro vzdálený monitoring. Jednak totiž plná verze běžící bez časového omezení je placená, druhá nevýhoda je vysoké zatížení procesoru WinConu při spuštění této aplikace. To se sice neprojevuje na řídicích algoritmech, jelikož jejich procesy mají vyšší prioritu než tento, ale například grafické prostředí na připojeném panelu k WinConu je pak značně zpomaleno.

## 4. Modelové úlohy vzdáleného řízení a vizualizace

V následující kapitole budou rozebrány vhodné modelové případy pro řídicí systém REX ve spojení s PAC WinCon, dále bude na realizované modelové úloze demonstrováno použití systému REX s komunikací do Matlabu a vizualizačních nástrojů jako je Promotic respektive Indusoft Web Studio. Vybraná modelová úloha bude realizovat řízení polohy reálné pístnice hydromotoru.

### 4.1 Výběr vhodných modelových případů pro systém WinCon + REX

Aplikace řízená pomocí zařízení WinCon a řídicího systému REX je vodná pro řízení středně rychlých strojů a procesů. Jde o ekonomicky poměrně výhodnou aplikaci, kdy vizualizační programové vybavení třetích stran může být provozováno na téže zařízení WinCon, kde již běží samotný řídicí systém. Přitom samotný řídicí systém REX je mnohem levnější, než konkurenční software IsaGraf. Přitom jej lze využít u „hard real-time“ aplikací s přesnými požadavky na odezvu systému a s dodržením přesné periody vzorkování. Minimální perioda spouštění úloh je v takovém případě 2 milisekundy, u obvyklých úloh bývá vhodná 5 – 10 milisekund, přičemž v samotném REXu existují diagnostické nástroje, jimiž lze jak měřit dobu výpočtu jednoho cyklu procesu, tak diagnostikovat opoždění spuštění řízeného procesu od zadaného času.

Další podstatná výhoda této konfigurace spočívá v kompatibilitě s programem Matlab – Simulink. Díky tomu je umožněno simulovat veškeré řídicí algoritmy REXu již před samotným nahráním tohoto algoritmu do cílového zařízení. V případě pomalejších řízených procesů (s odezvou řádově 100 milisekund) je dokonce možné řídicí schéma v cílovém zařízení (tedy ve WinConu) napojit na schéma v Simulinku a odtud provádět samotné prvotní řízení, včetně úprav různých parametrů během řízení.

Vhodná aplikace pro dříve zmíněný řídicí systém se ukázala modelová úloha řízení polohy pístnice hydromotoru. Tato aplikace je dostatečně rychlá na to, aby se zde mohly být demonstrovány rychlostní limity daného řídicího systému, ale zároveň i při nastavení pomalejší periody spouštění řídicího schématu, je rychlost odezvy pro tuto úlohu zcela dostatečná.

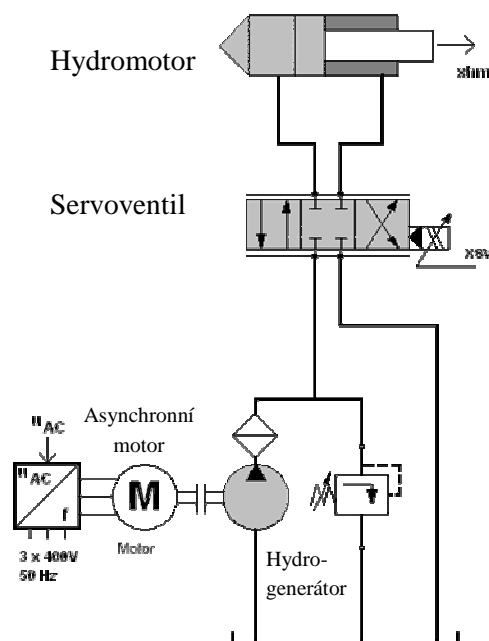
Systém REX na WinConu umožňuje připojení libovolné vizualizační aplikace přes vestavěný OPC server. Typická perioda vyčítání hodnot z OPC serverů se obvykle pohybuje mezi 100 milisekundami až několika sekundami či minutami. Díky tomu budou průběhy zobrazených technologických veličin ve vizualizačních aplikacích zobrazovány s mnohem menší periodou, než je perioda spouštění daného schématu v řídicím systému REX. Zároveň přímo řídicí systém REX umožňuje záznam historických trendů do paměti RAM a jejich následnou archivaci případně vyčítání a zobrazování. Záznam trendů těchto historických trendů je možné provádět s periodou stejnou jako je perioda spouštění odpovídajícího schématu v REXu.

## 4.2 Popis zvolené řízené technologie

Modelová úloha sestává z řízení polohy pístnice hydromotoru (viz Obr. 12). Systémový tlak oleje z hydrogenerátoru je usměrňován pomocí čtyřcestného ventilu do dvou protiběžných komor pístnice. Jednotlivé vstupně-výstupní signály řízené technologie jsou připojeny k vstupně-výstupním zásuvným modulům v základní jednotce WinCon 8739.

Ve WinConu byly použity dva paralelní zásuvné moduly I-8017 s analogovými vstupy ve slotech 2 a 3. Dále sériový modul I-87024 ve slotu 5 se čtyřmi analogovými výstupy. Nastavení polohy dvoucestného ventilu je prováděno přivedením napětí v rozsahu -10V až 10V, nastavení otáček motoru pomocí výstupu 0V až 10V do měniče. Zpětná vazba čidla polohy je v rozsahu 0V až 10V. Dále bylo možné měřit tlaky v obou komorách pístnice, systémový tlak, zpětnou vazbu skutečné polohy ventilu a otáčky motoru hydrogenerátoru.

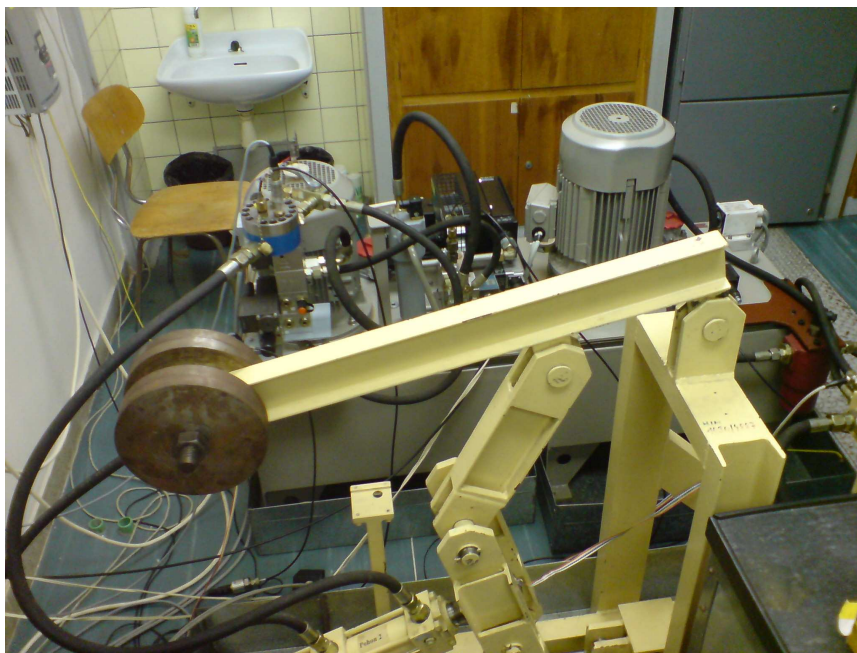
Řídicí algoritmus měl být navržen pro běh v systému REX ve WinConu přičemž pro vizualizaci zvoleného řízeného procesu byl zvolen jednak software pro PC - Promotic a Matlab-Simulink a následně software InduSoft Web Studio pro vizualizaci běžící přímo na WinConu a to buď na připojeném monitoru nebo formou internetových stránek z web-serveru (viz Obr. 13).



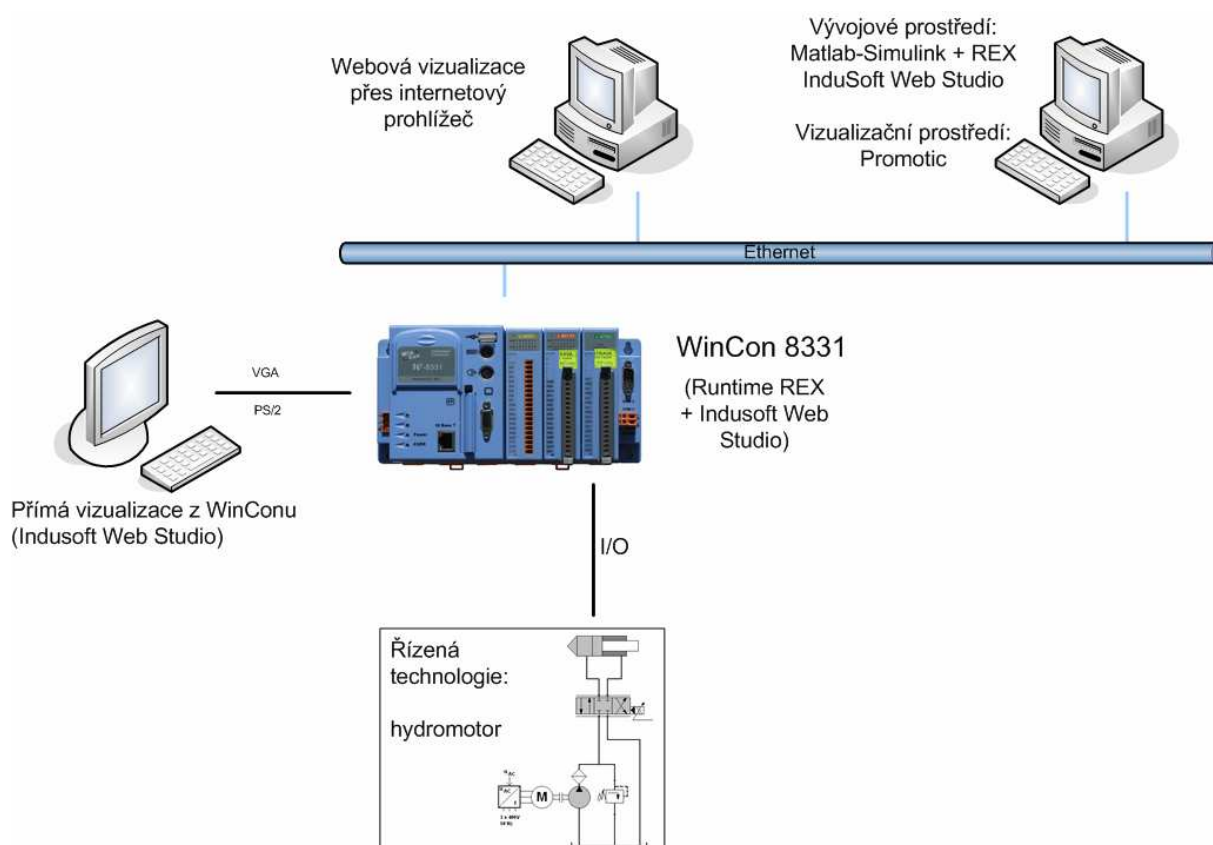
Obr. 11: Schéma řízené technologie

Pořadí	WinCon -I/O modul	Port na WinConu	Signál - zkratka	Signál	Rozsah	Fyzikální rozměr
1	I-8017 (1.)	Vin1	pA	Tlak v komoře A	(-10) - 10 V	0 – 21 MPa
2	I-8017 (1.)	Vin2	pS	Systémový tlak	(-10) - 10 V	0 – 21 MPa
3	I-8017 (1.)	Vin3	xhm	Poloha hydromotoru	0-10 V	0 – 0,12 [m]
4	I-8017 (1.)	Vin4	pB	Tlak v komoře B	(-10) - 10 V	0 – 21 MPa
5	I-8017 (1.)	Vin7	xsv	Poloha servoventilu	(-10) - 10 V	
6	I-8017 (1.)	Vin-	GND	Signálová zem		
7	I-8017 (2.)	Vin2	nAC	Skutečné otáčky motoru	0-10 V	0 – 1450 [ot/min]
8	I-8017 (2.)	Vin-	GND	Signálová zem		
9	I-87024	Vout1	u + u1/A	Akční zásah servoventilu	(-10) - 10 V	
10	I-87024	Vout2	uAC	Nastavení otáček motoru	0-10 V	0 – 1450 [ot/min]

Tabulka 9: Seznam použitých technologických vstupů a výstupů



Obr. 12: Pohled na řízenou technologii



Obr. 13: Schéma struktury navrhovaného řídicího systému s PAC WinCon, systémem REX a vizualizacemi v podobě software Promotic, InduSoft Web Studio a webově-orientované vizualizace přes internetový prohlížeč

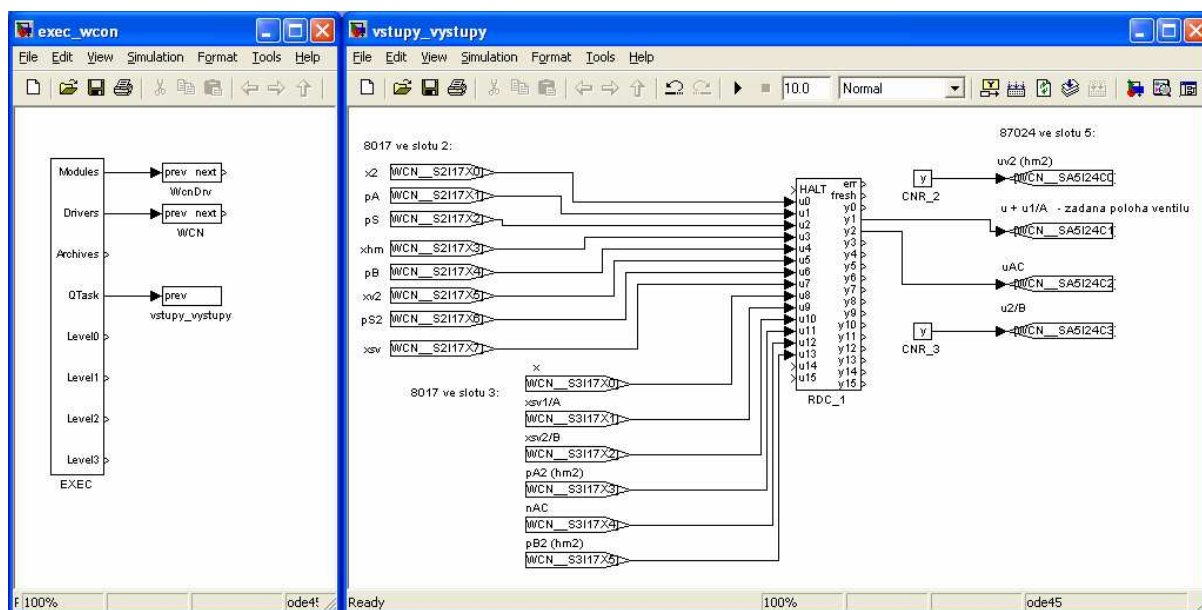
### 4.3 Propojení řídicí aplikace v REXu se schématem v Simulinku

Následující kapitola popisuje návrh a propojení řídicího algoritmu REXu a Matlab-Simulinku. Komunikace obou prostředí byla zajištěna prostřednictvím bloku „RDC“ z knihovny REXLib. Pro demonstraci řízení zmíněným způsobem byly vytvořeny dvě rozdílné aplikace. První realizuje samotné řízení prostřednictvím schématu běžícího v Simulinku, kdy byly blokem RDC přenášeny akční zásahy do REXu ve WinConu a zároveň vyčítány regulované hodnoty z REXu zpět do Simulinku. Schéma v ŘS REX ve WinConu zde slouží pouze pro čtení a zápis hodnot analogových vstupů a jejich odesílání periodické do schématu Simulinku. Druhá aplikace provádí řízení aplikované přímo do schématu REXu ve WinConu, přičemž Simulink zde pouze zobrazuje naměřené průběhy.

#### 4.3.1 Řízení z Matlab-Simulinku

V první řadě bylo vytvořeno řídicí schéma pro řízení z Matlab-Simulinku. Pomocí bloku RDC jsou jednotlivé hodnoty procesních veličin přenášeny přes ethernetové rozhraní do bloku RDC v REXu na WinConu a nastavovány na výstup nebo vyčítány ze vstupů a vráceny zpět do Simulinku.

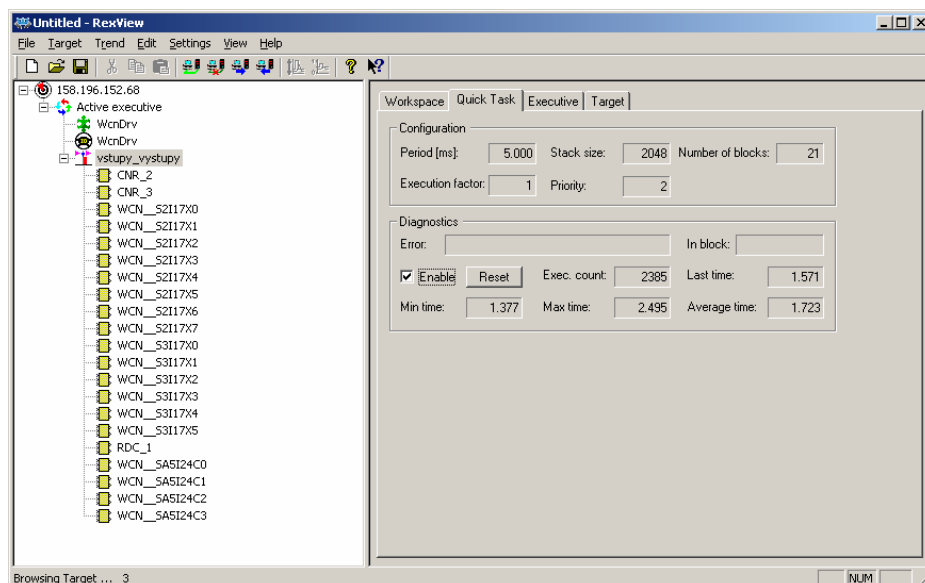
Program v REXu se skládá z hlavní části EXEC\_WCON obsahující konfigurační blok EXEC (viz Obr. 14) a příslušné subrutiny s názvem „vstupy\_vystupy“. V subrutině je definováno čtení hodnot vstupů a zápis hodnot na výstup. Subrutina je definována na úrovni QTask – tedy úloze s vysokou prioritou a periodou spouštění nastavenou v tomto případě na 5ms.



Obr. 14: Schéma pro zápis a čtení hodnot technologických vstupů a výstupů REXu ve WinConu včetně komunikačního bloku RDC

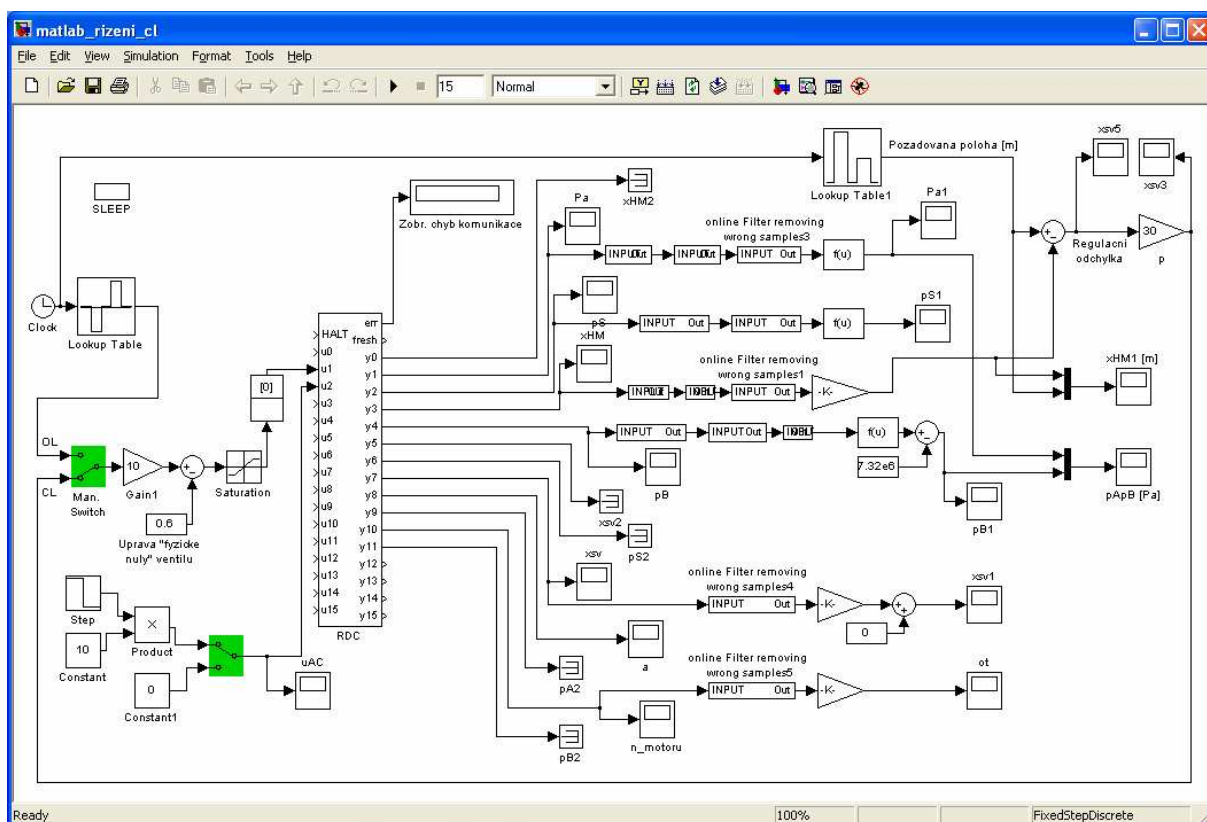
Pomocí nástroje REXView je možné zjistit dobu provádění jednotlivých subrutin (viz Obr. 15). To vyžaduje vytvoření spojení k REXu ve WinConu zadáním jeho IP adresy, označení požadované subrutiny a v záložce „Quick Task“ zapnutí diagnostiky. Z jednotlivých údajů je zřejmé, že průměrná doba provádění této rutiny je 1.723 milisekund, maximální 2.495 milisekund.





Obr. 15: Diagnostika úlohy v nástroji REXView

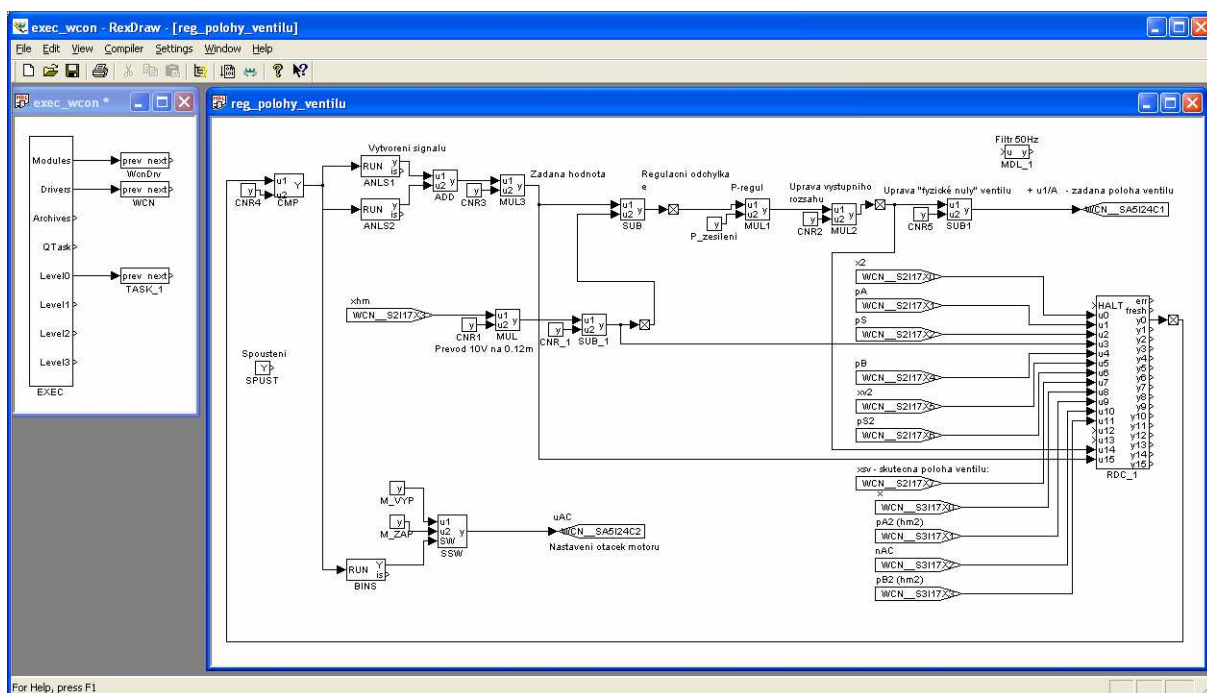
Program umožňuje zpětnovazební řízení polohy s P-regulátorem. Při spuštění zapne motor a provede celou sekvenční řadu požadovaných poloh pístnice. Pomocí bloku RDC program přenáší hodnoty nastavovaných proměnných do REXu a zároveň tímto způsobem hodnoty z REXu vyčítá. Perioda spouštění schématu definovaného v bloku „SLEEP“ byla nastavena na 100 milisekund.



Obr. 16: Blokové schéma v Simulinku se zpětnovazebním obvodem a sběrem a zobrazením technologických veličin na PC včetně komunikačního bloku RDC

### 4.3.2 Řízení z REXu ve WinConu a zobrazování průběhů v Matlabu

Celý řídicí algoritmus je aplikován v REXu ve WinConu a Matlab-Simulink přes blok RDC pouze zobrazuje naměřené průběhy, případně spouští sekvenci několika skokových změn požadované polohy pístnice. Propojení mezi PC a WinConem bylo realizováno přes ethernetové rozhraní na školní síti. Schéma řídicího algoritmu v REXu – viz Obr. 17. Perioda spouštění dané řídicí úlohy byla nastavena na 10 milisekund. Vlevo nahoře ve schématu se nacházejí dva bloky „ANLS“. Ty slouží pro generování impulsů dané šířky a doby trvání a lze říci, že generují požadovanou polohu pístnice. Skutečná poloha je měřena z analogového vstupu zásuvného modulu a odečítána od žádané, čímž vzniká regulační odchylka. Tuto zpracovává P-regulátor. Dále jsou v řídicí smyčce schématu bloky pro úpravu akční veličiny, úpravu „fyzické nuly“ ventilu a výstup na daný analogový výstup zásuvného modulu. Vpravo ve schématu se nachází blok „RDC“, který provádí komunikaci s odpovídajícím si blokem v RDC na jiném počítači. Blok umožňuje přenášet až 16 proměnných typu double. V tomto případě jsou přenášeny všechny hodnoty procesních veličin řízené technologie.

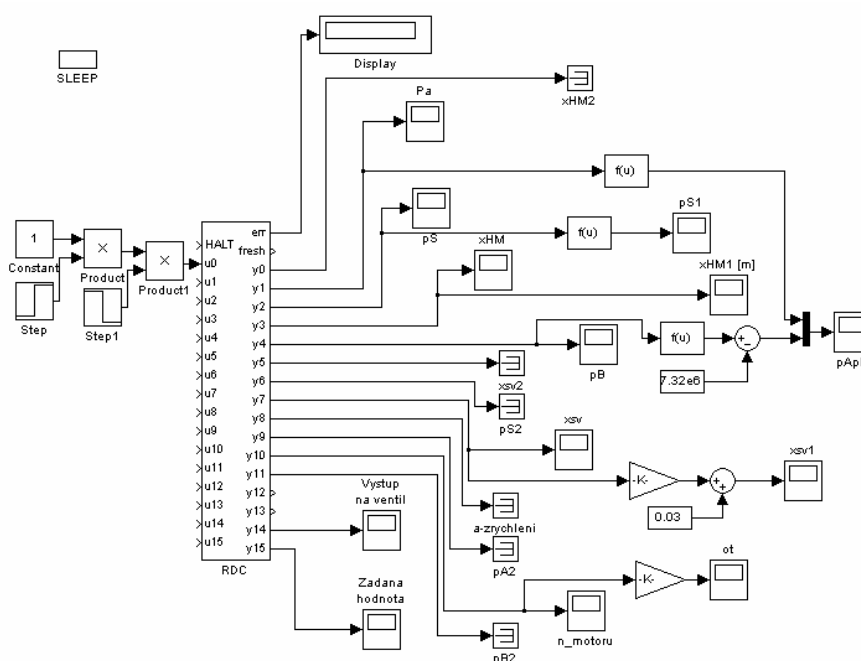


Obr. 17: Blokové schéma zpětnovazebního obvodu v REXu z aplikace REXView (pro cílové zařízení WinCon) s komunikačním blokem RDC, který zde slouží pro přenos procesních dat do Simulinku na PC

Program v Simulinku pak především zobrazuje průběhy daných procesních proměnných, přičemž zároveň upravuje jejich rozsahy do odpovídajících fyzikálních rozměrů. Po spuštění této úlohy se zároveň odešle impuls s délkou 10 sekund do REXu, kterému slouží jako synchronizační puls spouštění sekvence žádaných poloh pístnice (viz Obr. 18).

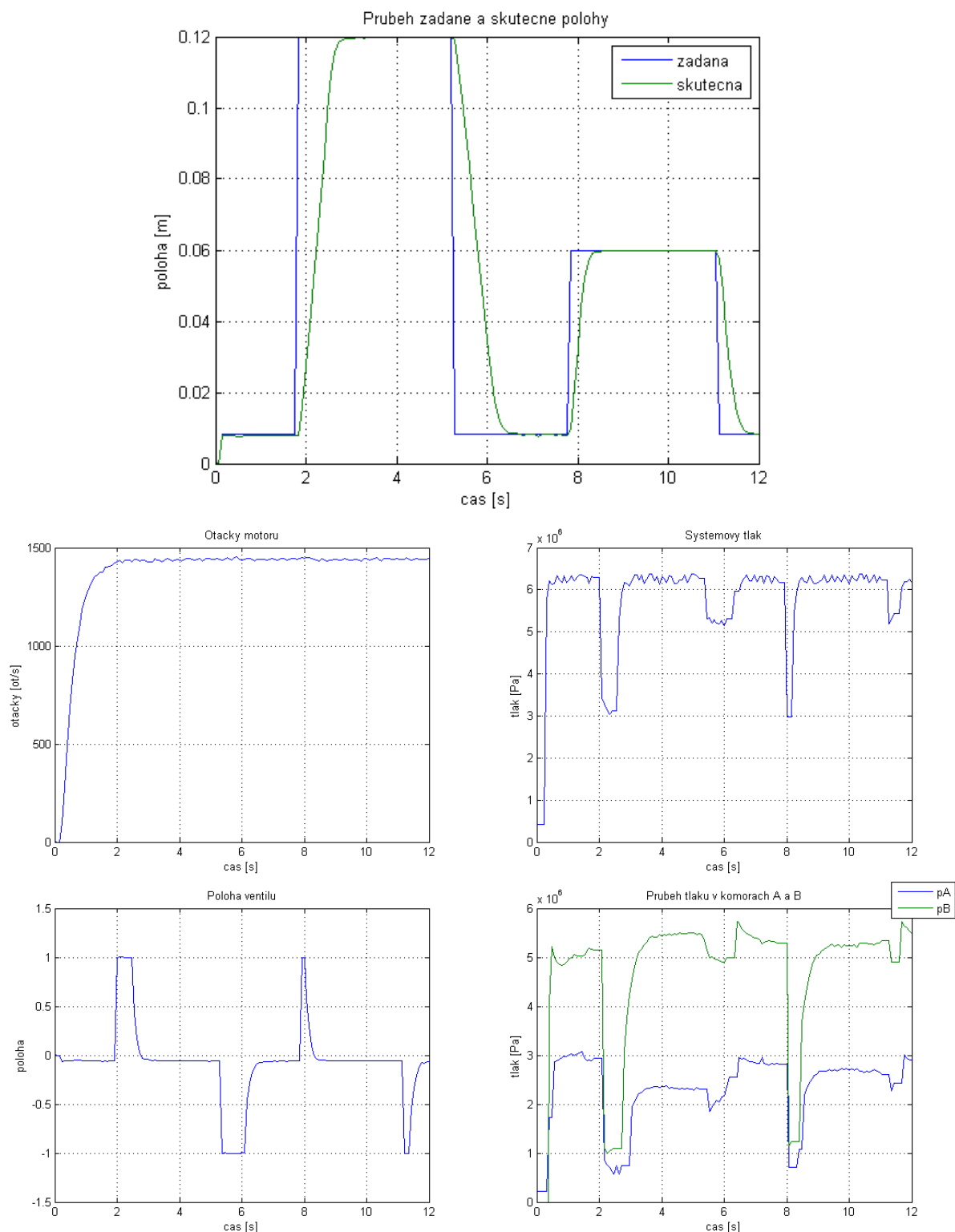
Abychom „synchronizovali“ běh simulace v Simulinku a řídicího algoritmu v REXu, bylo nutné do schématu v simulinku vložit blok „SLEEP“ z REX Library, který by měl zaručit cyklické spouštění úlohy v daném intervalu. Navíc je třeba nastavit v nastavení simulace v „Simulation -> Configuration Parameters-> Fixed Step Size“ na periodu spouštění stejnou jakou má blok „SLEEP“. REX Controls doporučuje nastavovat hodnotu parametru „SLEEP“ alespoň na 100ms. V poslední verzi 1.41 však vzhledem ke stále rostoucímu výpočetnímu výkonu nových PC byly povoleny i menší časy. Proto byly v rámci diplomové práce vyzkoušeny i nižší periody spouštění, při kterých se však již vyskytovaly problémy s měřením v „reálném čase“.

Simulink na PC nebyl schopen vykonávat úlohu v daném vyčleněném čase a docházelo ke zpoždění měřené úlohy od reálného času, i proto, že v tomto případě nelze využít Real-time možností programu Matlab jeho toolboxů. Na měřených grafech bylo patrné že si měřené průběhy odpovídají s průběhy žádanými, přičemž ale jejich časové osy nerespektují průběh v real-time a jsou zpožděné, posunuté v čase. Proto bylo provedeno měření skutečné doby běhu celé sekvence pomocí funkce „etime“. Výsledkem je, že doba provádění celého měření je závislá na parametru SLEEP a žádná nižší hodnota než 100 ms nepřinesla správnou délku doby její realizace. Při řízení z programového prostředí Matlab/Simulink a nastavení parametru „SLEEP“ na 20ms se pak celé úloha prováděla namísto žádaných 15 sekund celých 23.4 sekund. Při nastavení na 80 ms byla doba 15 sekund vykonávána 17.5 sekundy. Při nastavení parametru na 100 ms již byla doba provádění správná.



**Obr. 18:** Schéma v Simulinku pro sběr procesních dat z REXCore běžící na WinConu při s využitím komunikačního bloku RDC. Vlevo nahoře se nachází blok „SLEEP“ zajišťující přesně definované periodické spouštění celého schématu

Jednotlivé průběhy procesních technologických veličin jsou zobrazeny na následujících v grafech (viz Obr. 19):



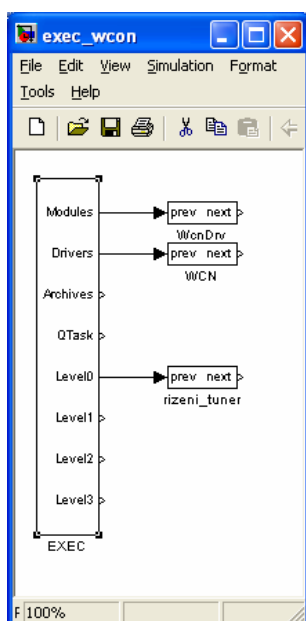
**Obr. 19: Průběhy vybraných technologických veličin - zleva nahoře: průběh žádané a skutečné polohy pístnice, otáčky motoru, systémový tlak, průběh tlaků v obou komorách pístnice, skutečná poloha ventilu. Komunikace schématu v Matlabu s REXem ve WinConu pře blok RDC**

## 4.4 Řídicí schéma REXu pro připojení různých vizualizačních aplikací

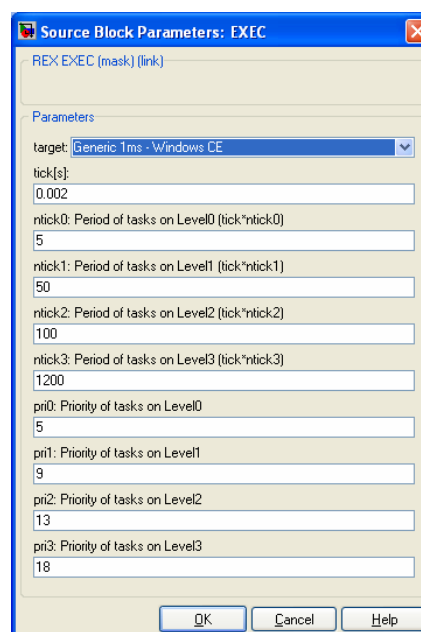
V prostředí Matlab-Simulink bylo navržena následující řídicí aplikace pro řízení modelové úlohy řízení polohy pístnice hydromotoru. Umožňuje spouštění regulace polohy buďto v uzavřené zpětnovazební smyčce nebo pouze jako ovládání polohy servoventilu – tedy otevřená regulační smyčka. V prvním případě je poloha řízena pomocí PID regulátoru s integrovaným momentových autotunerem firmy RexControls. K této aplikaci je možné připojit libovolný vizualizační software, který podporuje standard OPC. Tato kapitola tedy popisuje pouze řídicí schéma, jednotlivé vizualizační aplikace jak pro WinCon tak PC jsou popsány až v následujících kapitolách.

### 4.4.1 Návrh řídicího schématu reálné úlohy

Každá řídicí aplikace systému REX se skládá alespoň ze dvou schémat .mdl. První schéma nazvěme tzv. konfigurační (viz Obr. 21). Vždy obsahuje blok „EXEC“, ve kterém jsou nastaveny parametry jak cílového zařízení – tedy v tomto případě Windows CE, tak periody spouštění řídicích úloh na jednotlivých úrovních a jejich priority (viz Obr. 20). Tyto úrovně rozlišuje ŘS REX čtyři, přičemž každé lze přiřadit jinou periodu spouštění i prioritu. Perioda spouštění se vypočte jako násobek základního taktu „tick“ REXu, který je udáván v milisekundách a vynásobeným parametrem „ntick“. V tomto případě byl zvolen základní takt 2 milisekundy a spuštění úlohy při každém pátém taktu, tedy každých 10 milisekund. Priority jednotlivých úloh není obvyklé nutně měnit. K bloku „EXEC“ jsou dále na výstupu připojeny bloky typu „Modules“ a jim odpovídající ovladače na výstupu „Drivers“, které zde obstarávají komunikaci s I/O moduly WinConu. K výstupu „Level0“ je připojen blok s názvem schématu, ve kterém je řídicí schéma –mdl. Řídicí schéma již neobsahuje žádné konfigurační bloky, pouze bloky pro samotnou regulaci. Výstup „QTask“ slouží pro spouštění úloh velmi rychlých úlohou s periodou stejnou základnímu taktu REXu.



Obr. 21: Konfigurační schéma REXu s blokem "EXEC"

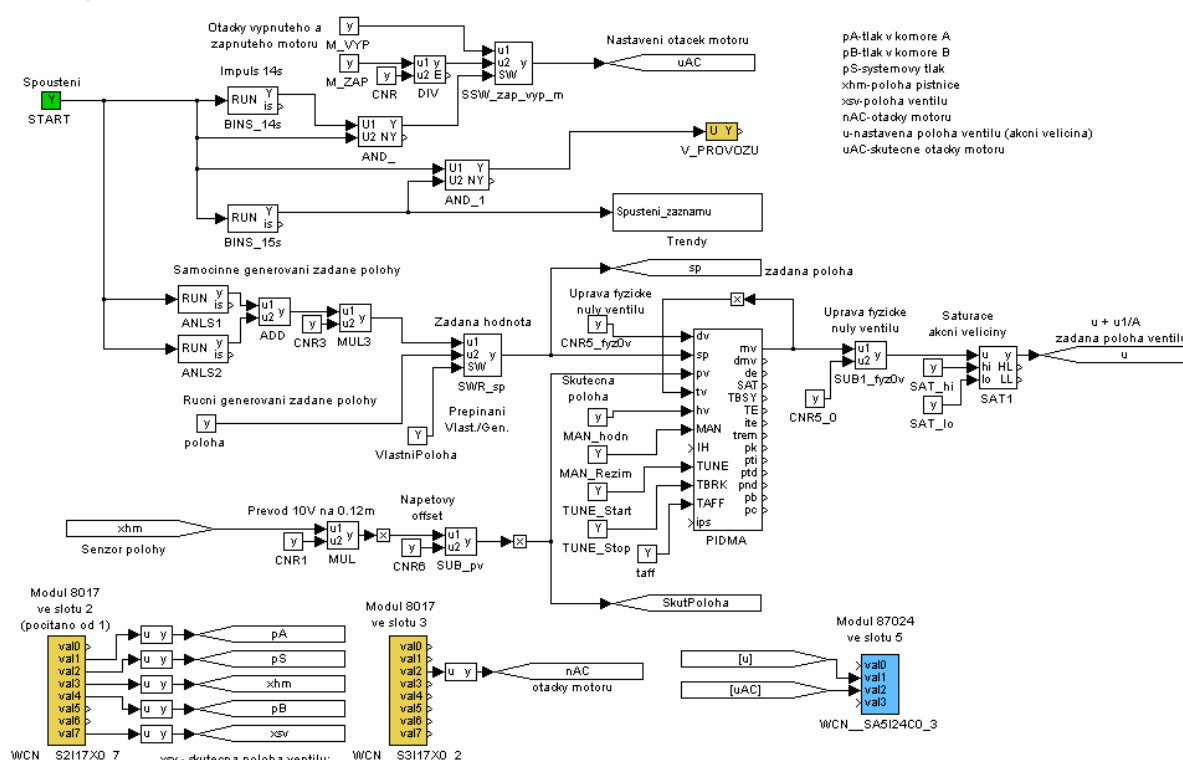


Obr. 20: Nastavení jednotlivých vnitřních parametrů bloku "EXEC"

Řídicí schéma je na Obr. 22. Spouštění řízení se provádí zápisem hodnoty „TRUE“ do binárního proměnné v bloku „START“. V tom případě se spustí asynchronní motor na žádané otáčky, což je definováno v horní části schématu. Zároveň se spustí záznam trendů všech sledovaných hodnot technologických veličin. Pokud se binární proměnná bloku „START“ změní na „FALSE“, motor i záznam trendů se okamžitě zastaví. Zároveň se regulace samočinně zastavuje po uplynutí doby patnácti sekund od startu.

K regulaci polohy pístnice slouží blok PIDMA, který realizuje PID regulátor s momentových auto-tunerem. Regulátor může fungovat v automatickém režimu, pokud je na vstupu „MAN“ hodnota typu „FALSE“. V takovém režimu řídí klasicky dle stanovených parametrů v tomto bloku ve zpětnovazební smyčce. V případě, že je na vstupu „MAN“ hodnota typu „TRUE“, regulátor pracuje v manuálním režimu, kdy hodnotu na vstupu „sp“ přenáší přímo na výstup „mv“. Žádaná poloha v automatickém režimu může být přepínána mezi konstantní hodnotou a samočinně generovanou sekvencí poloh.

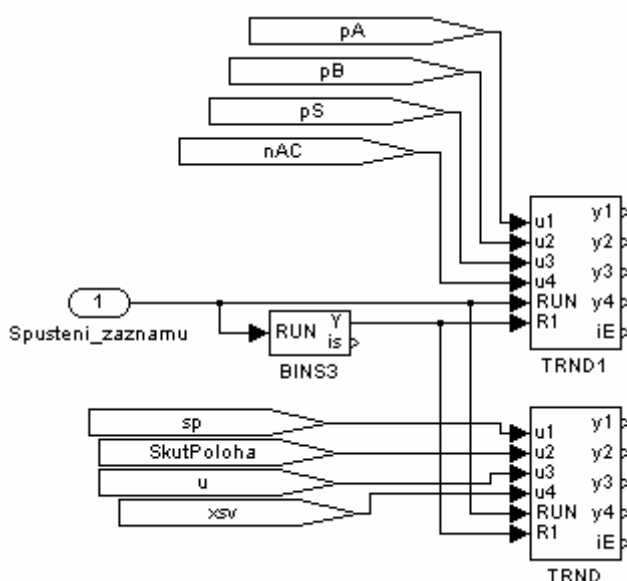
V dolní části schématu se nacházejí bloky pro přístup k I/O modulům ve WinConu. Žlutou barvu mají bloky modulů analogových vstupů. Název bloku vychází z konvence REXU pro pojmenovávání I/O bloků. V tomto případě název bloku „WCN\_\_S2I17X0\_7“ označuje I/O modul WinConu ve druhé slotu, typ modulu je I-8017 (v názvu bloku je obsaženo poslední dvojčíslí typu modulu) a napěťový rozsah -10 až 10 voltů na vstupech 0 až 7. Druhý I/O blok je konfigurován podobně, pouze slot jemu odpovídajícího modulu je 3. Modře označený blok zapisuje hodnoty na analogové výstupy modulu I-87024, konvence pro pojmenovávání je obdobná, pouze je upravena pro jiný sériový typ modulu.



Obr. 22: Hlavní řídicí schéma v systému REX

Ve schématu je využíván systém „GOTO“ a „FROM“ vlajek, které zajišťují propojení jednotlivých částí algoritmu bez viditelného propojení vodiči díky čemuž zjednodušují a zpřehledňují celé schéma. Zároveň umožňují propojovat různé úlohy REXu případně proměnné v různých subsystémech. Příklad bloku této vlajky je „xhm“, která je připojena k danému výstupu bloku I/O modulu a představuje skutečnou polohu pístnice. S touto vlajkou se dále pracuje jak ve stejném schématu, kdy je použita jako regulovaná veličina na vstupu PID regulátoru, tak v subsystému „Trendy“ pro záznam do trendů

Blok „Trendy“ slouží pro záznam jednotlivých technologických veličin do trendů v paměti RAM ve WinConu. Blok je vytvořen jako subsystém z dalších bloků, s náběžnou hranou na vstupním parametru typu boolean se nejdříve provede vymazání všech trendů, dále pak při stále hodnotě

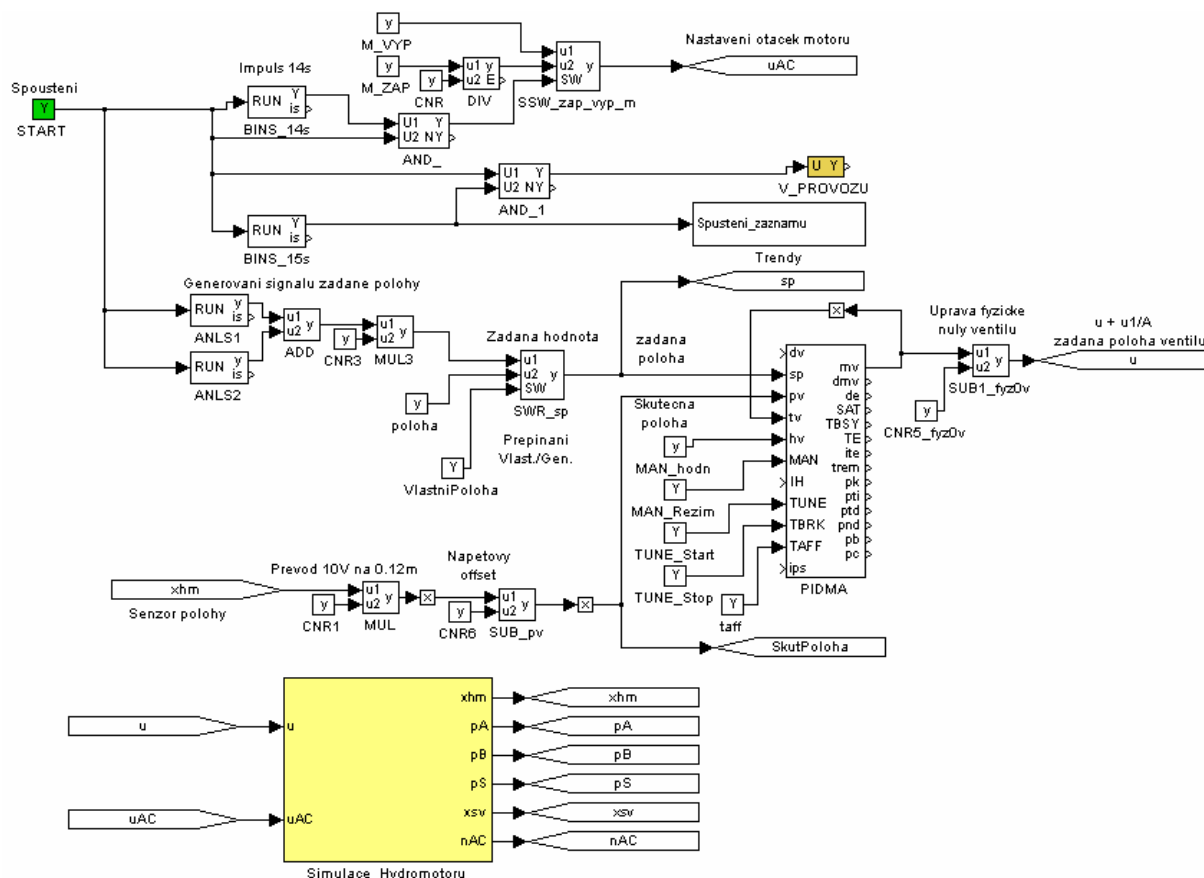


Obr. 23: Vnitřní schéma subsystému pro záznam sledovaných technologických veličin do trendů

„TRUE“ na vstupu probíhá ukládání. Vnitřní schéma subsystému je na Obr. 23. Tendovány jsou tyto veličiny: tlaky v obou komorách pístnice, systémový tlak, žádané otáčky motoru, skutečná a žádaná poloha pístnice a skutečná a žádaná poloha servoventilu. Vlastní ukládání hodnot je realizováno dvěma bloky „TRND“ z knihovny REXLib. Nastavení vnitřních parametrů těchto bloků je stejné, pouze mají připojeny jiné sledované veličiny. V parametrech těchto bloků je možné nastavovat četnost ukládání do paměti vůči taktu spouštění – zde bylo nastaveno při každém spuštění úlohy, dále typ ukládaných dat (všechny zvoleny jako „Doble“) a velikost zásobníku pro tyto data, po jehož zaplnění se starší hodnoty začnou přepisovat novými.

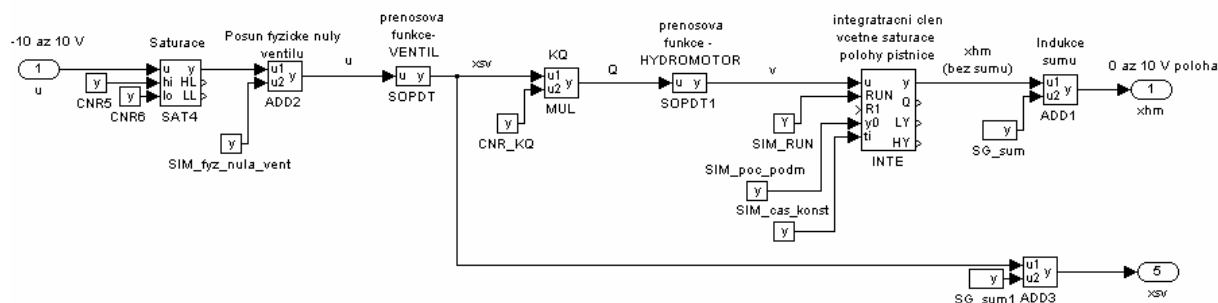
#### 4.4.2 Návrh simulačního řídicího schématu pro model soustavy

V případě vytváření řídicího algoritmu, ovšem za předpokladu, že známe alespoň základní matematický model řízené soustavy, je možné testovat chování takového řídicího schématu i včetně připojené vizualizační aplikace, a to bez nutnosti připojení reálné soustavy. Díky tomu, že řídicí systém REX umožňuje jednoduše realizovat simulování reálných soustav, bylo vytvořeno i druhé řídicí schéma bez bloků s ovladači I/O karet, přičemž jednotlivé I/O vlajky pro přístup k I/O měřicím modulům byly nahrazeny subsystémem simulující právě danou technologii (viz Obr. 24). V tomto subsystému je model astatické soustavy hydromotoru realizován pomocí integračního bloku a bloků přenosových funkcí z knihovny REXLib. Protože popisované řídicí schéma neobsahuje žádné I/O vlajky, je možné jej spouštět a testovat na libovolné platformě podporované řídicím systémem REX, tedy nejen ve WinConu, ale i na PC. Takové simulační schéma dané technologie může být přesným matematickým modelem řízené úlohy, což však není nutnou podmínkou. Simulační schéma především umožňuje jednoduše navrhovat a testovat vyvíjenou vizualizační aplikaci, která bude komunikovat s danou exekutivou REXu a to bez nutnosti připojení ke skutečné reálné modelové úloze prostřednictvím například WinConu.



Obr. 24: Hlavní řídicí schéma s modelem řízené soustavy

Subsystem *Simulace\_Hydromotoru* zde realizuje zjednodušený matematický model řízené soustavy. Vnitřní struktura subsystému je složena především z bloků přenosových funkcí *SOPDT*, bloku integrátoru *INTE* a bloků *SG* generující šum. Část vnitřní struktury subsystému *Simulace\_Hydromotoru*, která realizuje model chování pístnice hydromotoru je zřejmá z Obr. 25. [16]

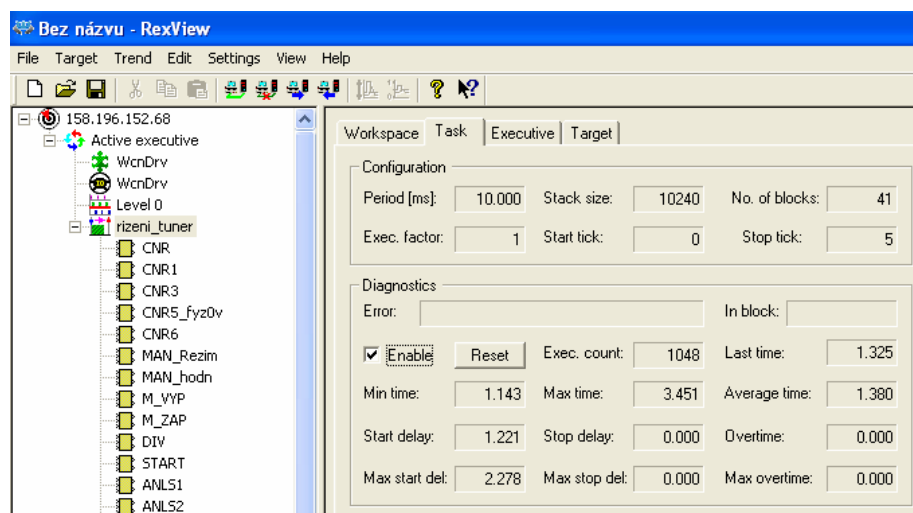


Obr. 25: Část vnitřní struktury subsystému *Simulace\_hydromotoru* představující zjednodušený matematický model řízené technologie – polohu hydromotoru



### 4.4.3 Diagnostika schématu prováděného exekutivou REXu

Nahrání řídicího schématu ze Simulinku do exekutivy REXu lze provádět přímo z nabídky daného konfiguračního schématu a zadání IP adresy cílového zařízení (WinConu). V tomto případě bude řídicí schéma v exekutivě REXu uloženo pouze do jejího prvního restartu. Pro trvalé uložení v exekutivě REXu je nutné nahrání přes aplikaci REXView, potom se již řídicí schéma spouští samočinně při každém startu exekutivy. Zároveň je možné přes REXView provádět diagnostiku (viz Obr. 26).



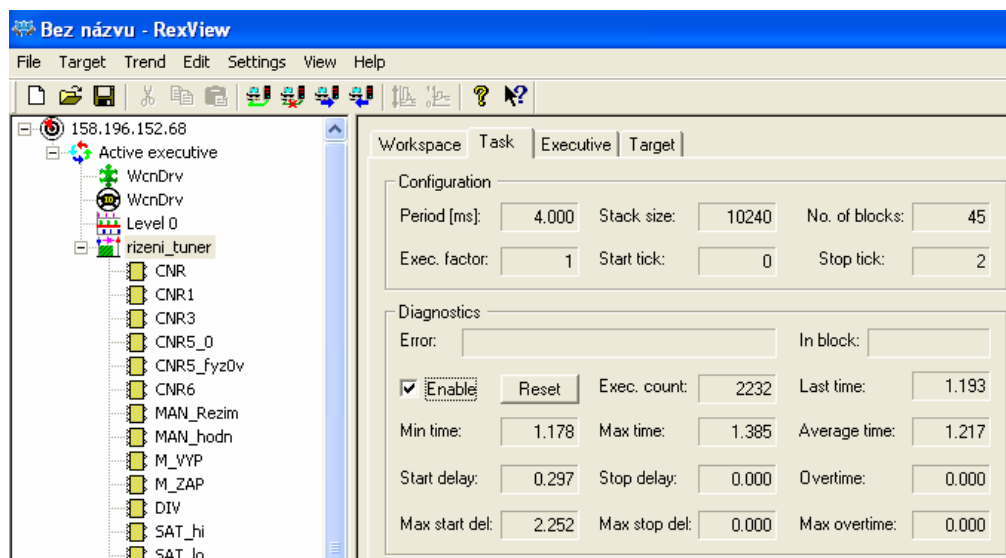
**Obr. 26:** Výřez z obrazu aplikace REXView sloužící i pro diagnostiku nahraného řídicího schématu (v tomto případě je diagnostikováno schéma pro řízení reálné úlohy s periodou spouštění 10 milisekund)

V tomto případě byla aplikace připojena k exekutivě ve WinConu se schématem pro řízení polohy hydromotoru, přičemž byla prováděna a průběžně zobrazována diagnostika úlohy. Odtud v části „Configuration“ je patrná jednak perioda spouštění schématu – 10 milisekund (*Period*), počet bloků (*No. Of Blocks*), násobek spouštění úlohy vůči spouštění bloku „EXEC“ (*Exec. factor*), počáteční takt spouštění úlohy a takt do kterého musí být daná úloha dokončena (*Start tick* a *Stop tick*). V části „Diagnostics“ jsou dále zobrazeny například údaje o délce výpočtu jednoho cyklu úlohy – doba výpočtu posledního cyklu byla 1.325 milisekund (*Last time*), průměrná doba 1.38 milisekund (*Average time*) a maximální 3.451 milisekund (*Max time*). Dále je zde zobrazeno zpoždění začátku výpočtu (*Start delay*), překročení povolené doby výpočtu (*Stop delay*) a mnohé další.

Přesný popis těchto parametrů a způsob jejich výpočtu však nejsou v příručce k REXu podrobněji popsány, tudíž lze jen předpokládat, že jednotlivé časové údaje zde budou odvozeny od základního taktu procesoru.

Z daných výsledků a časů provádění jednotlivých cyklů aplikace v exekutivě REX je zřejmé, že by bylo možné snížit periodu spouštění této úlohy. Protože byl nastaven základní takt systému REX na 2 milisekundy a jednotlivé úlohy je možné spouštět pouze v časech s násobkem tohoto taktu, byla následně zvolena perioda spouštění každé 4 milisekundy. Při této periodě se jednotlivé diagnostikované doby provádění prakticky neliší od předchozí vyšší periody spouštění, přičemž stále

maximální přednastavené doby provádění úlohy nebyly překročeny. Proto lze vyvodit závěr, že perioda spouštění této úlohy může být nastavena i na 4 milisekundy.

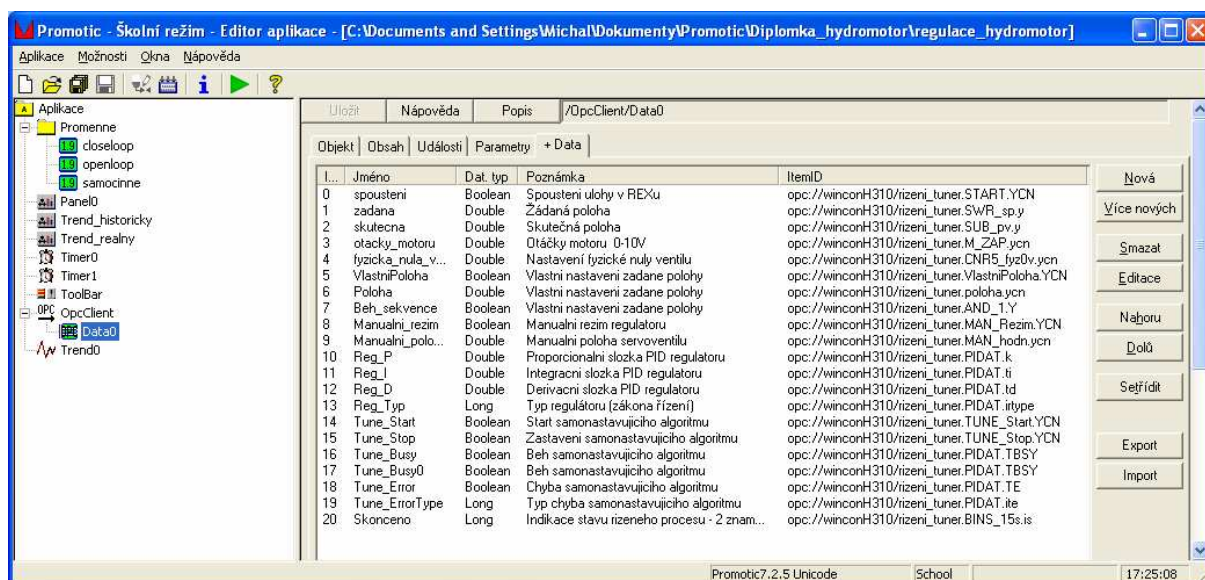


Obr. 27: Výřez z okna aplikace REXView a diagnostiky spouštěného řídicího schématu reálné úlohy s periodou 4 milisekundy

## 4.5 Vizualizační aplikace v prostředí Promotic

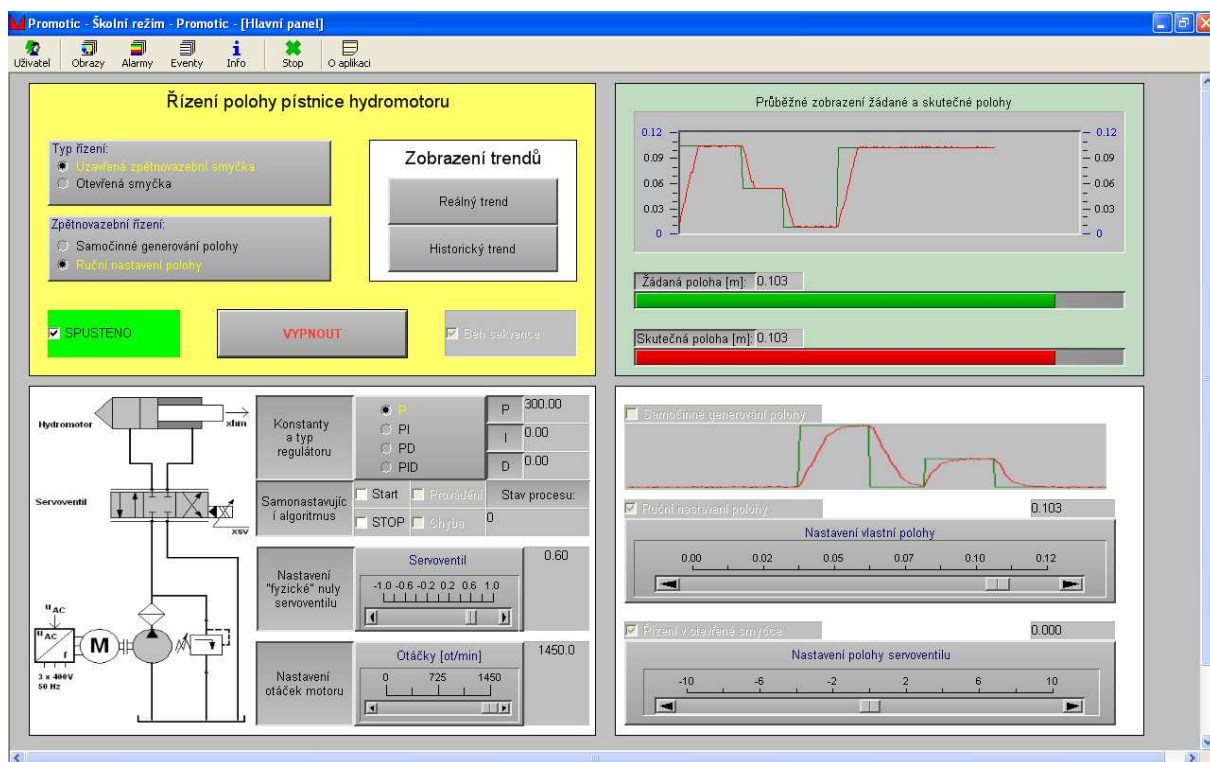
Vizualizační aplikace v prostředí Promotic byla vytvořena pro snadné ovládání řídicího algoritmu REXu popsaného v předchozí kapitole 4.4.1 a to z PC připojeného přes školní ethernetovou síť k WinConu. Samotná komunikace probíhá přes OPC Server REXu.

OPC Server řídicího systému REX, který běží na Windows CE neumožňuje přístup z jiného zařízení na síti. Toto omezení však lze obejít instalací REXu na PC, na němž bude provozována vizualizační aplikace, v tomto případě Promotic a poté nakonfigurování cílového zařízení OPC serveru na PC tak, aby odposlouchával exekutivu REX na jiném, než lokálním zařízení. Tento postup byl již popsán v kapitole 3.4.7. Soupis všech použitých proměnných je na Obr. 28. Jednotlivé ID prvků v OPC Serveru není nutné znát, jelikož jak OPC Server tak Promotic podporují vyhledávání a výběr jakýchkoliv prvků ze stromové struktury OPS standardu.

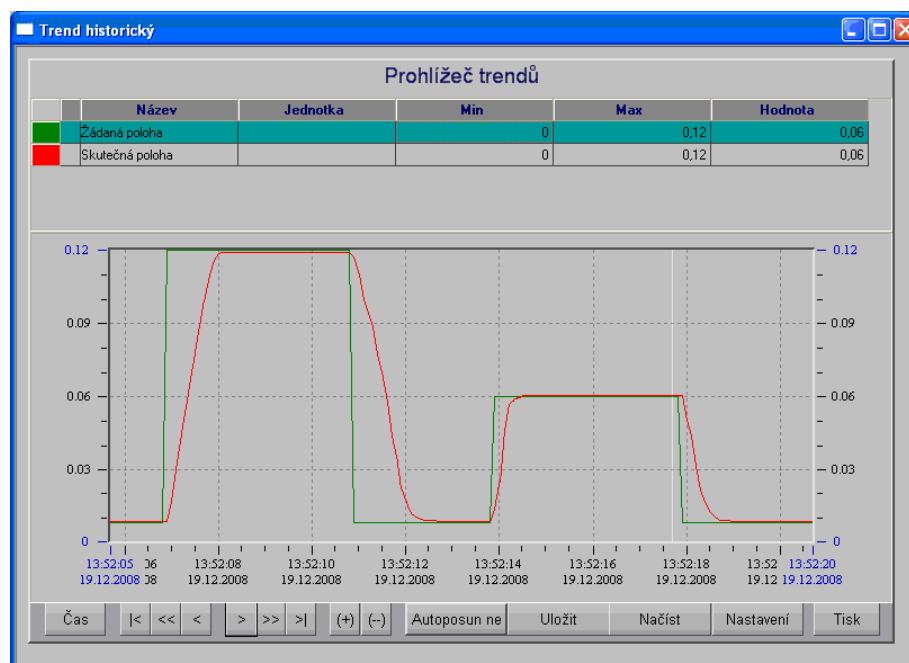


Obr. 28: Struktura aplikace v prostředí Promotic a seznam všech prvků pro iteraci z OPC Serveru REXu

Aplikace umožňuje snadné spouštění řídicího algoritmu v REXu ve WinConu (viz Obr. 29). Tlačítkem „ZAPNOUT“ se spustí celá řízená technologie, přitom se funkce tlačítka změní na funkci odstavení celé technologie. Aplikace umožňuje záznam do historických trendů několika technologických veličin. V pravé části se nachází reálný trend skutečné a žádané polohy a „progressbar“ se stejnou funkcí. Požadovanou polohu písnice lze v automatickém režimu zadávat ručně, nebo samočinně generovat již přednastavenou sekvencí několika žádaných poloh. V manuálním režimu je možné nastavovat přímo polohu servoventilu, který reguluje pohyb a polohu písnice hydromotoru. Při řízení v automatickém režimu (tj. ve zpětnovazební smyčce) lze průběžně měnit konstanty PID regulátoru, stejně tak lze spustit samonastavující algoritmus, který využívá momentový autotuner z řídicího systému REXu. Taktéž je z vizualizace umožněno nastavení „fyzické nuly“ servoventilu a požadovaných otáček asynchronního motoru.



Obr. 29: Vzhled vizualizační aplikace z prostředí Promotic na PC, která slouží pro ovládání řídicího algoritmu REXu běžícího ve WinConu při řízení polohy pístnice hydromotoru



Obr. 30: Příklad historického trendu v prostředí Promotic

## 4.6 Vizualizační aplikace v prostředí Indusoft Web Studio

Ve vizualizačním nástroji Indusoft Web Studio byla vytvořena druhá aplikace aplikace pro snadné spouštění a ovládání řídicího algoritmu v REXu. Výsledná aplikace běží přímo na WinConu a je zobrazována na připojeném monitoru, odkud je možné ji ovládat počítačovou myší a klávesnicí. Zároveň tato aplikace funguje jako webový server, je možné se k ní připojit z libovolného webového prohlížeče z počítače na síti. V obou případech je zobrazován stejný řídicí panel.

Vyvíjená aplikace poběží na WinConu přičemž bude komunikovat přes lokální OPC Server s exekutivou REXu. K tomuto OPC Serveru však není při návrhu vizualizační aplikace přístup, takže není možno jednoduše „browsovat“ jeho jednotlivé položky. Nicméně tuto „složitost“ je možné odstranit. Vzhledem k tomu, že InProc OPC Server běžící na WinConu se projevuje stejně, jako tento OPC Server spuštěný na PC, je možné do lokální PC-exekutivy nahrát stejné regulační schéma jako poběží ve WinConu, pouze s tím rozdílem, že nebude obsahovat žádné bloky I/O modulů a bude mít lehce změněné nastavení bloku „EXEC“. Tento způsob umožňuje nastavit připojení v IWS takovým způsobem, že bude fungovat jak na lokálním PC tak i po nahrání přímo do WinConu.

Konkrétně bylo u OPC Serveru REXu použito v parametru „TargetName“ zkratka „@“, tedy OPC Server komunikoval s REX exekutivou přes sdílenou paměť. Tato volba musela být samozřejmě nastavena i na lokálním vývojovém PC.

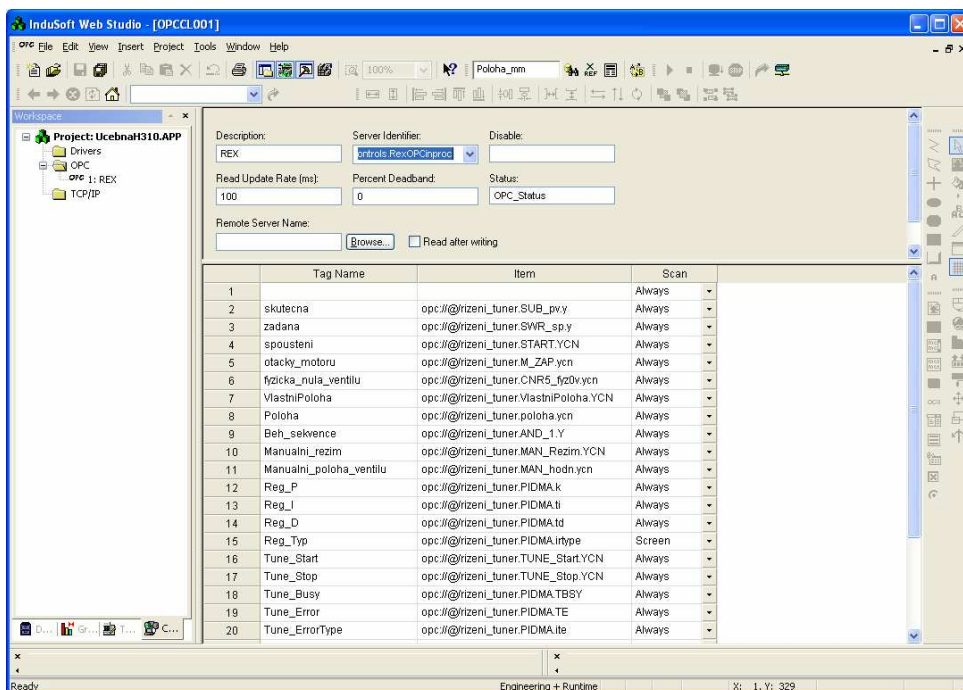
Aby bylo možné do cílového zařízení nahrát aplikaci z IWS, je nutné aby na něm již běžel runtime IWS. Potom se export této aplikace do daného zařízení provede položkou z menu vývojovém prostředí IWS. IWS tu výhodu, že pro různá cílová zařízení z pohledu operačního systému je návrh těchto aplikací zcela stejný, pouze se v menu IWS přepne na jiné cílové zařízení. Navrženou aplikaci lze zároveň uložit jako html stránku, která komunikuje s runtime IWS a distribuuje tak tuto aplikaci do webových prohlížečů na ethernetové síti.

WinCon existuje ve verzích buďto bez licence runtime IWS nebo již tuto licenci obsahuje, v takovém případě je již součástí obrazu operačního systému. Runtime IWS ve WinConu je přeinstalován, včetně vzorové demonstrační aplikace, zároveň, pokud není nainstalována licence, běží runtime IWS v demonstračním módu cca 2 hodiny. V případě, že je nutné runtime IWS nově nainstalovat, provede se následující postup. Z instalačního adresáře IWS na PC se ze složky „BIN/REDIST/WINCE41/Armv4/Bin“ zkopíruje soubor „CeView.exe“ na paměťovou kartu WinConu, odkud se následně spustí. Nyní je již runtime IWS připraven k připojení z vývojového prostředí IWS na PC, kde z nabídky v menu provedeme instalaci systémových souborů (do WinConu) a následně je možné již vytvořenou aplikaci taktéž z nabídky v menu „Project-> Execution Environment“ do WinConu nahrát. Pokud je nutné zajistit, aby se IWS spouštělo automaticky po startu operačního systému na WinConu, je nutné tuto aplikaci zaregistrovat přes „WinCon Utility“.

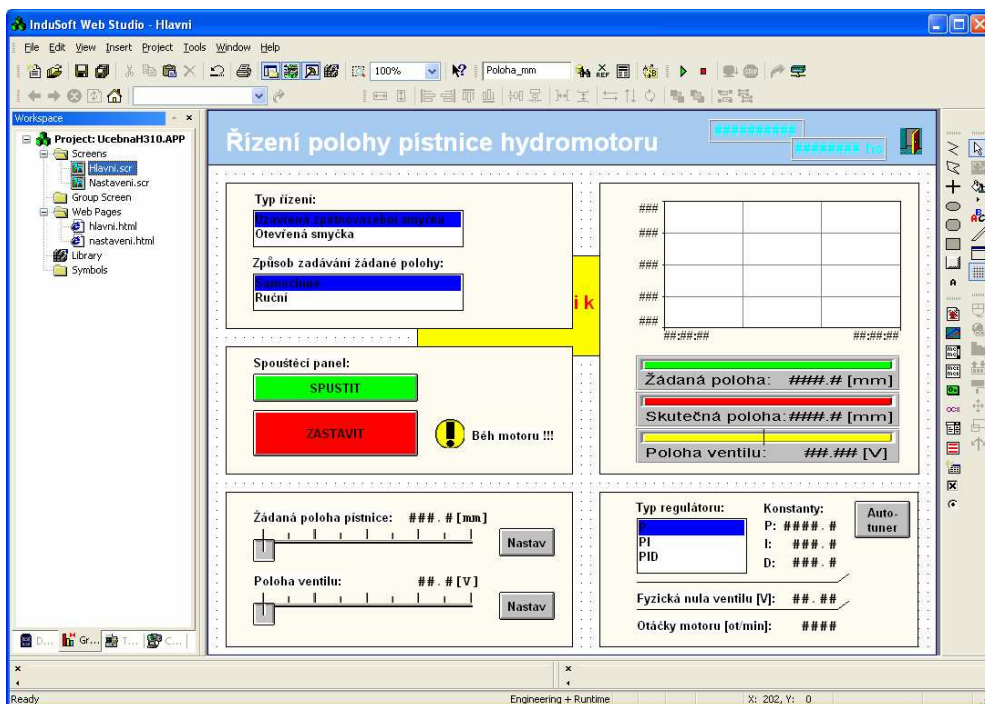
IWS zároveň umožňuje vytvářet webové orientované vizualizační aplikace. V tomto případě, jakmile je v IWS vytvořen hotový projekt, pomocí nabídky z menu „File -> Save as Html“ uložíme projekt do formy webových stránek, ty se budou nyní nacházet v adresáři, kde je celý projekt ukládán ve složce „/Web“. Celý obsah této složky je nutné ručně zkopírovat do HTTP adresáře ve WinConu. Kopírovat lze například pomocí flash-disku nebo nebo FTP serveru integrovaného ve WinConu. Nastavení cesty k souborům jak výchozího HTTP serveru tak FTP se provádí opět přes „WinCon Utility“. Webové stránky vytvořené přes IWS pak komunikují přímo s runtime IWS pomocí JavaAppletů. Spuštění této webové vizualizace v prohlížeči se provede zadáním adresy ve formátu:

„http://IP\_adresa\_WinConu/nazev\_hlavniho\_okna.html“, v tomto případě je konkrétní adresa tato: „http://158.196.152.68/hlavni.html“.

Navržená aplikace se skládá ze dvou oken. Okno s názvem hlavní slouží pro řízení polohy píštěnice přes REXexecutivu. Druhé okno „nastavení“ slouží pouze pro spuštění a ladění samonastavujícího algoritmu.

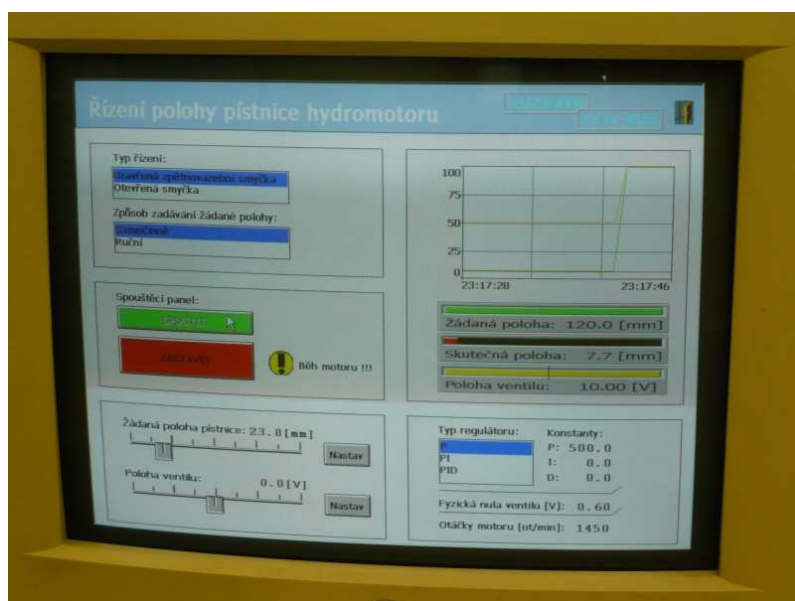


Obr. 31: Nastavení připojení k OPC Serveru REXu a jednotlivé nadefinované prvky

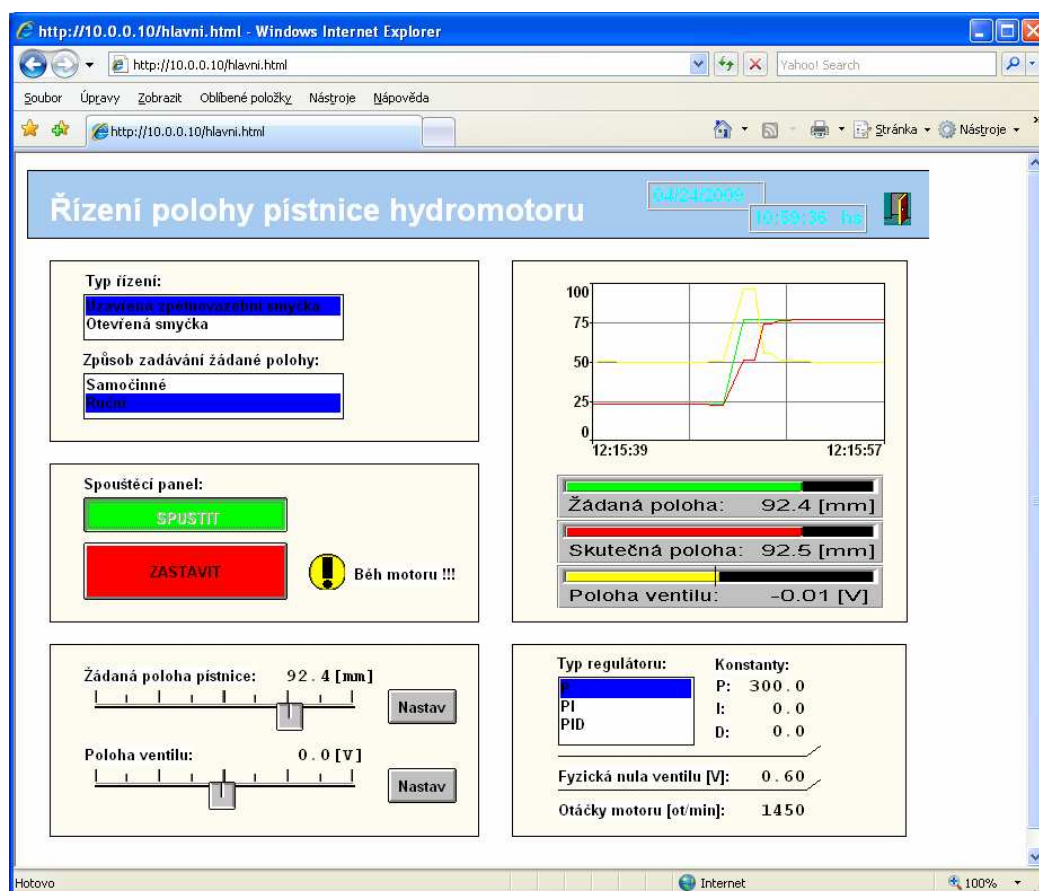


Obr. 32: Vzhled vizualizačního panelu v prostředí InduSoft Web Studio





Obr. 33: Navržená vizualizační aplikace na monitoru připojeném přímo k WinConu

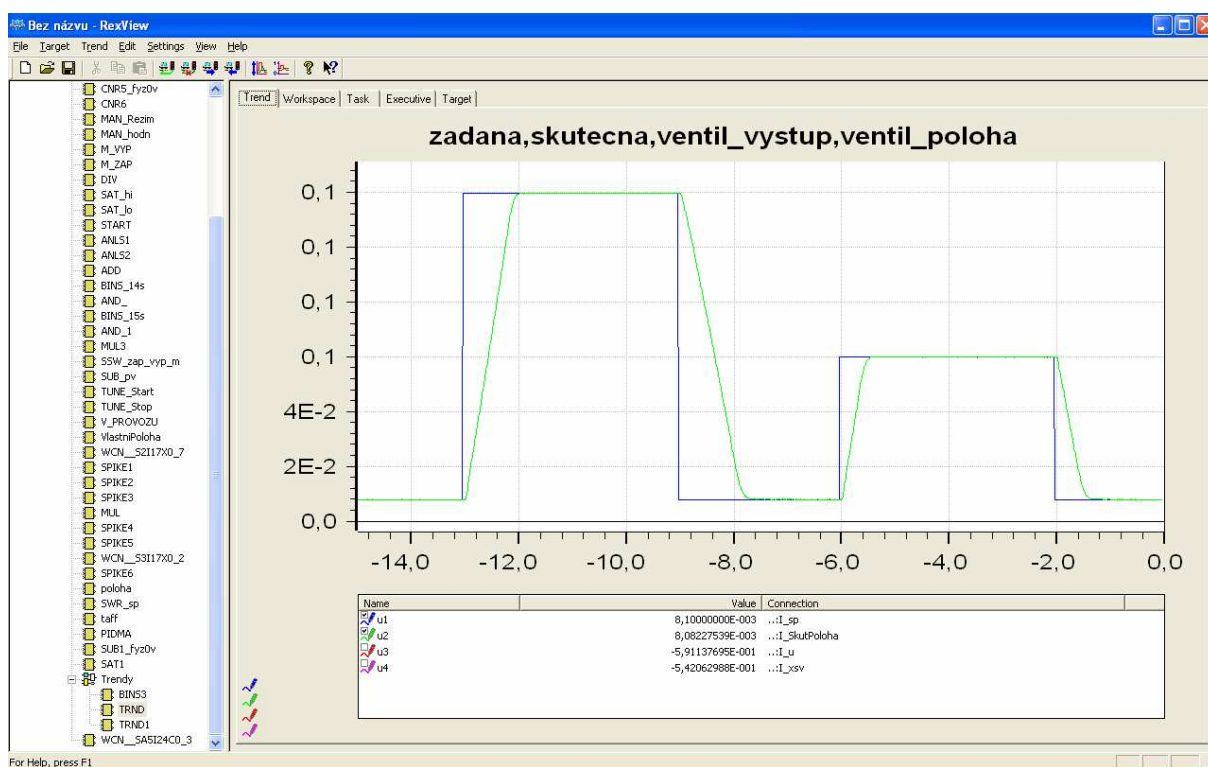


Obr. 34: Vzhled navržené vizualizační webové aplikace v prohlížeči MS Internet Explorer

## 4.7 Historické trendy vyčítané z aplikace REXView

Záznam historických trendů do paměti RAM WinConu vycházel z bloku „TRND“. Tímto způsobem uložená data lze následně vyčítat několika způsoby. První způsob využívá nástroje *REXView*, kde je možné jednotlivé průběhy zobrazovat a případně ukládat do textového souboru. Další možnost je pomocí nástroje *JavaTrend*, který slouží taktéž pro zobrazení a záznamu dat do souboru. Vyčítání dat je taktéž umožněno rozhraním *Automation*. V tomto případě se komunikuje přes rozhraní *.dll* knihovny *REXAutSv.dll*, tudíž je možný přenos dat jak do projektu ve Visual Studiu, tak i například do Excelu. V rámci diplomové práce však byl řešen pouze přenos a zobrazení dat v *REXView* a jejich následný export se zobrazením v Matlabu.

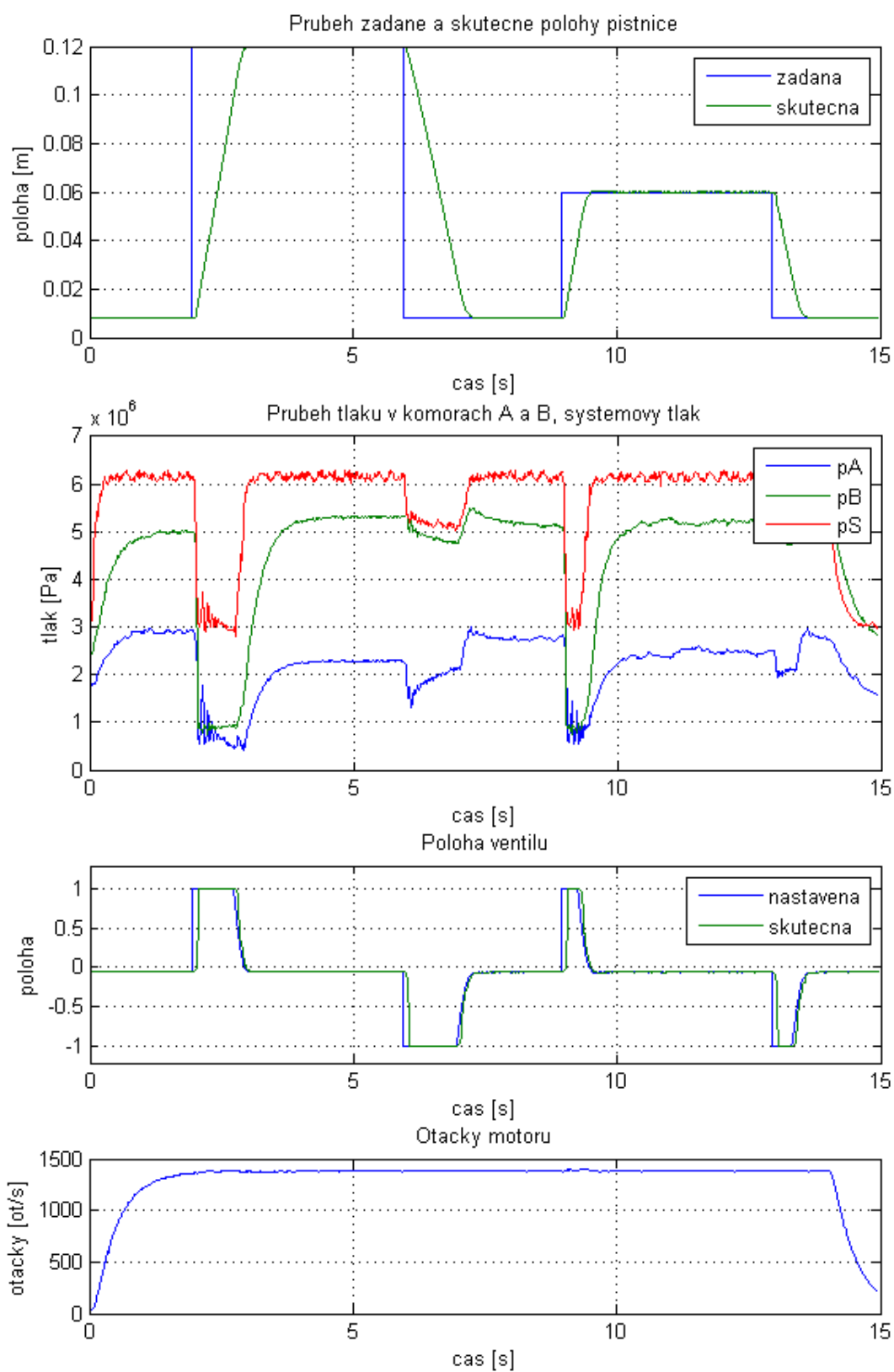
Následující naměřené průběhy byly získány řízením z vizualizace v IWS ve WinConu, přičemž řídicí schéma REXu z kapitoly 4.4.



Obr. 35: Zobrazení průběhu trendu v aplikaci REXView - žádané (modrý průběh) a skutečné polohy pístitnice (zelený průběh) při samočinném generování polohy

Obsah všech historických trendů z paměti RAM WinConu byl po měření a následném postprocesingovém načtení v aplikaci *REXView* exportován do *.csv* souboru, načten v Matlabu a byly zobrazeny jednotlivé průběhy měřených technologických veličin (viz Obr. 36).



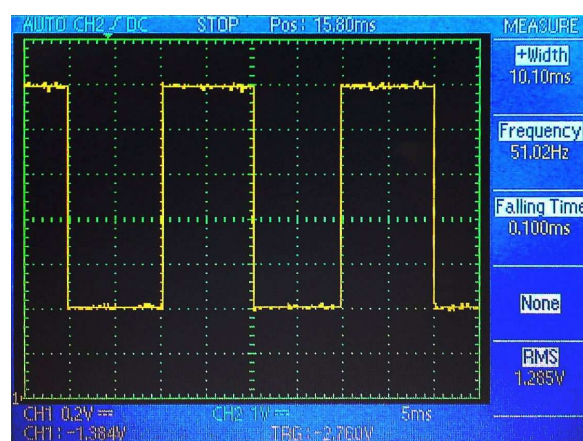


**Obr. 36:** Naměřené průběhy z historických trendů uložených v paměti RAM ve WinConu

U zobrazených průběhů je nutné si uvědomit, že všechny hodnoty analogových vstupů zásuvných modulů I-8017, z nichž některé průběhy vycházejí, musely být filtrovány bloky *SPIKE*. Filtrační bloky *SPIKE* zde sloužily k potlačení ohraničených chybových impulsů v měřeném signálu, které byly způsobeny softwarově a to vadným firmware v těchto zásuvných modulech. Jde o chyby vyskytující se pouze u některých zásuvných modulů se starším firmware, jenž v době prodeje ještě nebyl zcela odladěn. Firmware všech zásuvných modulů je možné přehrát jejich u dodavatele, do doby dokončení této práce však nebyl update jejich firmware proveden. Bloky *SPIKE* realizují nelineární filtr, kdy potlačují izolované chybové impulsy měřeného signálu, zároveň však při skokové změně měřeného signálu způsobují dopravní zpoždění. V důsledku toho je průběh například skutečné polohy ventilu při jeho skokové změně časově značně posunut vůči průběhu jeho nastavené polohy.

#### 4.8 Měření reálné periody spouštění řídicího schématu REXu

Měření periody pravidelného spouštění schématu REXu na WinConu mělo potvrdit nebo vyvrátit real-time vlastnosti dané platformy WinConu ve spojení s řídicím systémem REX. Základní myšlenkou měření bylo vytvořit ve schématu funkci, která by pravidelně přenastavovala buď digitální nebo i analogový výstup některého zásuvného modulu ve WinConu, a to s periodou, která by byla rovna alespoň dvojnásobku periody spouštění daného schématu v REXu (Shannon Kotelníkův teorém). Bohužel žádný z dostupných modulů digitálních výstupů, nebyl podporován ovladačem REXu pro I/O moduly. Tudíž byla vytvořena funkce, která v pravidelných okamžicích synchronizovaných právě se spouštěním daného schématu, měnila hodnotu analogového výstupu modulu I-87024 mezi dvěma zadanými napěťovými hladinami. Perioda spouštění schématu byla nastavena na 10 milisekund, přičemž jako řídicí schéma bylo použito schéma z kapitoly 4.4, navíc rozšířené o již zmíněnou funkci. Ta spočívala v programovacím bloku *REXLANG*, který umožňuje do řídicího schématu přidávat funkční kódy psané ve skriptovacím jazyce podobného kódu C. Výstupní průběh napětí analogového výstupu byl měřen digitálním osciloskopem se samočinným výpočtem šířky pulsů. Ačkoliv by šířka měřených pulsů měla být ideálně kolem 10 milisekund, v reálu se pohybovala přesně na 30 milisekundách. Odtud plyne hypotéza, že mezi procesem zápisu hodnoty analogového vstupu přes ovladač I/O karet v REXu a samotným nastavením této hodnoty na analogový výstup je vnořen proces, který periodicky provádí zápis žádaných hodnot na jednotlivé výstupy daného paralelního modulu, přičemž způsobuje pře-vzorkování žádaného výstupu. Zpoždění výstupu může souviset s komunikací samotného modulu po vnitřní sériové sběrnici s hlavním CPU. Reálné chování a průběh zápisu hodnoty na výstup však není popsán v manuálu REXu ani v manuálech k WinConu či konkrétním modulům, proto jej nelze blíže specifikovat. Z daného chování však lze vyvodit závěr, že nejnižší perioda obnovování analogové hodnoty na výstupu zásuvného modulu I-87024 je 30 milisekund. V případě zásuvného modulu I-8024, který komunikuje



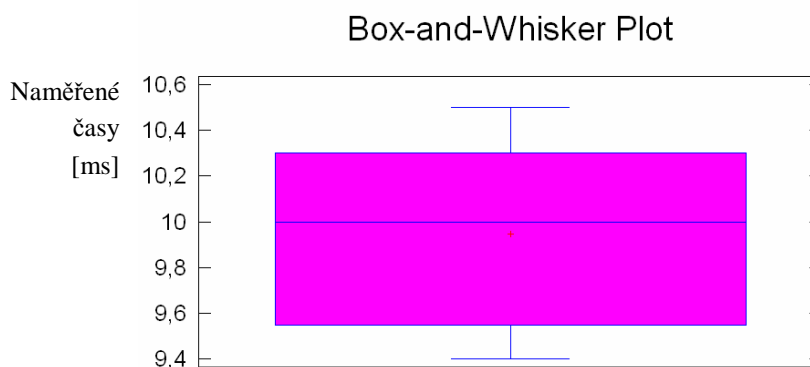
Obr. 37: Časový průběh generovaných impulsů analogového výstupu paralelního modulu I-8024 při periodě spouštění řídicího schématu 10 ms

s hlavním CPU WinConu přes vlastní interní sběrnici byla ověřena výsledná perioda zápisu na výstup 10 milisekund. Odtud je zřejmá rozdílná maximální frekvence zápisu na výstupy mezi paralelními moduly řady 8k (např. I-8024) a sériovými moduly řady 87k (I-87024).

Zároveň bylo provedeno měření šířky generovaný impulsů analogového zásuvného modulu I-8024 ve WinConu, pomocí digitálního osciloskopu EZ Digital DS-1080C. Šířka těchto pulsů odpovídá reálné periodě spouštění schématu REXu ve WinConu. Zde byly zjištěny následující statistické data: průměr 9,945 milisekund, medián 10 milisekund a směrodatná odchylka 0,37 milisekund (viz Tabulka 10 a Obr. 38). Tyto hodnoty se velmi souhlasí s diagnostikovanými hodnotami z aplikace REXView. Podrobnější popis měření a jednotlivé tabulky naměřených hodnot – viz Příloha II.

Počet vzorků měření	20
Průměr	9,945 [ms]
Medián	9,830 [ms]
Rozptyl	0,141 [ms]
Minimum	9,40 [ms]
Maximum	10,50 [ms]
Rozsah	1,1 [ms]
Směrodatná odchylka	0,37 [ms]
Standardní chyba	0,08 [ms]

**Tabulka 10: Statistika naměřených dat – ověření skutečné periody pravidelného spouštění schématu REXu ve WinConu při nastaveném parametru periody spouštění 10 milisekund**



**Obr. 38: Krabicový graf rozložení naměřených časů – reálné periody pravidelného spouštění schématu REXu ve WinConu při nastaveném parametru periody spouštění 10 milisekund**

## 4.9 Zhodnocení dosažených výsledků

V rámci diplomové práce byly nejprve podrobně nastudovány možnosti řídicího systému REX ve spojení s PAC WinCon a různými vizualizačními systémy. Následně byly tyto skutečnosti ověřeny realizací na skutečné modelové úloze pro řízení. Zhodnocení dosažených výsledků bude rozděleno na několik částí. První o samotném řídicím systému REX včetně vývoje v Matlabu, dále vizualizace na PC v Promotic a nakonec vizualizace přímo ve WinConu v prostředí InduSoft Web Studio.

V rámci diplomové práce bylo použito softwarových nástrojů následujících verzí:

- Řídicí systém REX verze 1.41
- Matlab verze 2007b a 2008b
- Promotic 7.2.5
- InduSoft Web Studio 6.0+SP5

Začneme zhodnocením řídicího systému REX. Zde bylo ověřeno, že středně složité řídicí schéma použité v modelové úloze lze ve WinConu spouštět s periodou 10 milisekund, přičemž pokud by na WinConu neběžel žádný další proces – například vizualizace v IWS, bylo by možné zkrátit spouštěcí periodu i na 4 milisekundy. Řídicí systém REX byl propojen s Matlab-Simulinkem pro měření technologických dat v reálném čase s využitím bloku RDC. To předpokládá blok RDC jednak ve schématu na straně WinConu, tak ve schématu Simulinku na PC. Rychlost přenosu těchto dat je však limitována především schématem na straně PC, kde pro korektní zobrazení dat v čase bylo nutné nastavit periodu spouštění schématu alespoň 100 milisekund. Prodloužení doby výpočtu na straně WinConu však není tak razantní, zde přidání bloku RDC představuje prodloužení výpočtu schématu cca o 1 milisekundu. V případě řídicího schématu v REXU bez bloku RDC (tedy schématu určenému pro připojení externích vizualizací) byla průměrná doba výpočtu jednoho cyklu úlohy řízení přibližně 1 milisekunda. Všeobecně lze říci, že blok RDC je použitelný jako náhrada komunikace přes standard OPC. V případě Matlabu je jeho nastavení i použití intuitivnější, než nastavení OPC klienta, a to i v případě, když daná verze Matlabu již obsahuje OPC Toolbox.

V následující fázi práce byla vytvořena vizualizační aplikace v prostředí Promotic pro ovládání zvolené modelové úlohy. Napojení Promoticu na PC k řídicímu schématu REXu ve WinConu se realizovalo připojením přes OPC Server REXu. V tomto případě byla ověřena perioda vyčítání dat z OPC Serveru 100 milisekund, což je v podstatě nejnižší rozumná perioda pro vyčítání dat pomocí komunikačního standardu OPC. Díky tomu jsou například i průběhy sledovaných trendových veličin jasně čitelné, na rozdíl od vizualizace v IWS, jak bude uvedeno dále.

Vytvoření vizualizace v prostředí IWS pro zařízení WinCon je poněkud složitější než v případě Promoticu. Výsledná aplikace je sice taktéž vyvíjena na PC, ale následně bude nahrána a spuštěna na cílovém zařízení – tedy WinConu. Při návrhu aplikace je nutné správně nastavit vyčítání dat z OPC Serveru ve WinConu, který však není ve vývojovém nástroji IWS přímo viditelný a tudíž bylo nutné jednotlivé parametry a vyčítané prvky nastavit tzv. „offline“, bez možnosti je pouze „vybírat“ ze seznamu dostupných.

Výsledná navržená aplikace sloužila pro ovládání řídicího schématu REXu dvěma způsoby. Zaprvé lokálně přímo pomocí monitoru, myši a klávesnice připojené přímo k WinConu. Zadruhé vzdáleně pomocí webového serveru ve WinConu, který vizualizaci realizuje formou internetových stránek a k němuž se lze připojit libovolným internetovým prohlížečem. Vizualizační okno má v obou případech stejný vzhled a funkčnost, rozdíl je však v rychlosti vykreslování trendů v runtime IWS.

V případě aplikace zobrazené na monitoru WinConu, nelze dosáhnout překreslování trendů rychleji, než jednou za sekundu, nastavení vyšší rychlosti obnovování systém IWS ignoruje. V případě webové aplikace se podařilo dosáhnout překreslování cca každých 200 milisekund, ovšem posun trendu byl již značně nepravidelný, proto se jako vhodná perioda nabízí opět alespoň jedna sekunda.

Runtime systému IWS běží ve WinConu na menší prioritě než procesy REXu. V důsledku toho při nastavení nižších spouštěcích period schématu REXu dochází ke zpomalení samotného uživatelského rozhraní IWS, tedy systém pomaleji reaguje na příkazy operátora a nepravidelněji pohybuje trendy. Při nastavení periody spouštění schématu REXu na 10 milisekund aplikace IWS nevykazovala žádné zpoždění vůči příkazům operátora, při nastavení periody 4 milisekundy však již rozhraní IWS vykazovalo značné zpomalení, stejně tak nestíhalo překreslovat trendy každou sekundu a místo toho probíhal posun trendu nepravidelně mezi dvěma až pěti sekundami.

Zároveň je nutné poukázat na nutnost použití filtračních bloků „SPIKE“ v řídicím schématu REXu. Při měření hodnot analogových vstupů WinConu na modulech I-8017 docházelo nepravidelně, přibližně jednou za sto vzorků k neměřené chybové hodnoty. Ty se projevovaly úzkými ohraničenými impulsy v naměřeném signálu. Po konzultacích jak s REX Controls tak prodejcem PAC WinCon, kdy obě společnosti se již s tímto problémem setkaly, se dospělo k závěru, že jde o problém způsobený chybou ve Firmware těchto měřicích modulů. Přehrání Firmware je však nutné provést u dodavatele a jelikož v době dokončování této práce nebyl stále ještě proveden, všechny hodnoty analogových vstupů vyžadovaly pro správnou funkci použití filtru.

Zároveň bylo provedeno měření pro odlišení vlastností modulů sériových a paralelních. Na základě vytvořené funkce, která v pravidelných okamžicích synchronizovaných právě se spouštěním daného řídicího schématu, měnila hodnotu analogového výstupu modulu mezi dvěma zadanými napětíovými hladinami. Perioda spouštění schématu byla nastavena na 10 milisekund. Testovány byly dva měřicí moduly – paralelní I-8024 a sériový I-87024. Výsledkem měření je zjištění, že paralelní modul provádí periodickou změnu výstupu v zadaném čase, na rozdíl od sériového modulu, kde je prováděn periodický zápis s periodou alespoň 30 milisekund. Při měření průměrné odchylky spouštěcí periody od žádané periody řídicího schématu REXu byly zjištěny následující statistiky: průměr 9,945 ms, medián 10 ms a směrodatná odchylka 0,37 ms.

PAC WinCon je vhodný pro řízení jak procesů typu HVAC, tedy například teploty – vytápění, ventilace, klimatizace, osvětlení v budovách tak pro řízení strojních technologií. Totéž platí při použití s řídicím systémem REX. V takovém případě je možné spouštět řídicí schémata generované ze Simulinku s periodou 10 milisekund, ovšem za předpokladu použití paralelních zásuvných měřicích modulů typu 8K. Sériové moduly 87K by v takovém případě prováděly akční zásahy periodicky co 30 milisekund.

<b><i>Limitní periody spouštění řídicího schématu REXu ve WinConu:</i></b>	
Perioda pravidelného spouštění řídicího schématu <b>REXu ve WinConu</b>	10 milisekund
Perioda pravidelného spouštění řídicího schématu <b>REXu ve WinConu</b> s bloky <b>RDC</b> pro komunikaci s Matlab-Simulinkem	10 milisekund
Perioda ukládání <b>historických trendů REXu</b> do paměti RAM <b>ve WinConu</b>	10 milisekund
<b><i>Limitní periody spouštění schématu v prostředí Matlab – Simulink na PC:</i></b>	
Perioda pravidelného spouštění schématu v <b>Simulinku</b> komunikujícího bloky <b>RDC</b> s exekutivou REXu ve WinConu	100 milisekund
<b><i>Limitní periody vizualizačních aplikací na PC:</i></b>	
<b>Promotic</b> - perioda vyčítání prvků ze schématu REXu ve WinConu při komunikaci přes REX <b>OPC</b> Server a ukládání do historických trendů	100 milisekund
InduSoft Web Studio – perioda vyčítání prvků ze schématu REXu ve WinConu při komunikaci přes REX <b>OPC</b> Server a ukládání do historických trendů	100 milisekund
<b><i>Limitní periody vizualizační aplikace ve WinConu:</i></b>	
<b>InduSoft Web Studio</b> – perioda vyčítání prvků z OPC serveru REXu a zobrazování trendů v případě <b>webově orientované aplikace</b> z web-serveru přímo ve WinConu a vizualizace v internetovém prohlížeči na PC	500 milisekund
<b>InduSoft Web Studio</b> – perioda vyčítání prvků z OPC serveru REXu a zobrazování trendů v případě aplikace běžící ve WinConu a provádějící vizualizaci <b>na připojeném monitoru</b>	1 sekunda

Tabulka 11: Přehled limitních nároků kladených na rychlost řízení s řídicím systémem REX

## 5. Závěr

Diplomová práce se zabývá možnostmi vzdáleného řízení a vizualizace procesů na SoftPLC WinCon 8000, dále demonstruje řízení na zvolené modelové úloze a definuje a ověřuje základní limity tohoto typu řízení. Jako základ popisovaného řídicího software sloužil řídicí systém REX a Matlab-Simulink. Zároveň zde existuje návaznost na bakalářskou práci (!!!odkaz!!!), kde byl návrh řídicích algoritmů zaměřen na vývojové prostředí MS Visual Studio, tudíž mohlo být provedeno porovnání výhod a nevýhod zcela odlišných způsobů návrhu těchto řídicích algoritmů.

Řídicí systém REX je poměrně nový produkt, jenž stále prochází vývojem. V době, kdy byla diplomová práce zadána, existoval REX v podobě s nesrovnatelně menším množstvím podporovaných jak cílových platform i ovladačů, tak popisů chování daného systému včetně vzorových příkladů. Některé návody nebyly zcela zřejmé a bylo často nutné dotazovat se přímo REX Controls. Přesto je nutné zdůraznit, že s každou aktualizací přibývá jak vhodně zpracované dokumentace tak ukázkových příkladů. Systému REX a jeho použití byly kromě této diplomové práce navíc věnovány dva semestrální projekty.

Vhodnou modelovou úlohou bylo zvoleno řízení polohy pístnice hydromotoru. V první řadě navržený algoritmus řízení umožňoval ladění řídicího schématu přímo za běhu aplikace v Simulinku. V tomto případě bylo demonstrováno propojení samotných systémů – Simulinku běžícího na PC a REXu ve WinConu. Ke komunikaci sloužily bloky RDC z knihovny funkčních bloků od REX Controls. Zároveň byla stanovena rychlost přenosu dat při daném typu komunikace.

V další části práce byl navržen řídicí algoritmus v REXu pro WinCon, jenž umožňoval připojení přes standard OPC k různým vizualizačním prostředím. Jako vhodný vizualizační nástroj bylo zvoleno Indusoft Web Studio, pro vizualizaci přímo ve WinConu a pro ovládání řízeného procesu jak z monitoru připojeného přímo k WinConu, tak pomocí integrovaného webového serveru – tedy formou internetových stránek. Druhý typ vizualizační aplikace v Promotic byl určen pro ovládání úlohy z PC. Výsledné porovnání všech typů vizualizací včetně komunikace blokem RDC bylo shrnuto v kapitole 4.9.

Následně provedené měření real-time odezvy přineslo jak výsledky periodicity spouštění schémat REXu ve WinConu, tak poukázalo na rozdílné reálné rychlosti měření a zápisu dat na měřicí zásuvné moduly. Výsledky naznačují, že při řízení systémem REX na platformě WinConu a dané složitosti řídicího schématu, lze dosahovat periody spouštění schémat 10 milisekund, přičemž však je nutno pracovat výhradně s paralelními měřicími moduly. Sériové moduly se projevovaly pomalejší odezvou, a to až 30 milisekund.

Vzhledem k nestálému vývoji nových PAC, především společnosti ICPAS, kdy jsou zvyšovány jak výpočetní tak komunikační schopnosti jednotlivých zařízení, je pravděpodobné, že si zařízení typu PAC budou ukrajovat větší podíl na trhu s PLC a různými kontroléry. Nejsou sice vhodné pro všechny typy úloh, ale existuje určitý okruh úloh, kde jejich výhody zcela převažují nad výhodami PLC.

## 6. Extended abstract

The diploma thesis describes possibilities of remote control and visualization on PAC WinCon 8000. The main part of the thesis deals with the system REX Controls that enables designing control algorithm for PAC in Matlab-Simulink. Usage of REX system is demonstrated in real application that controls the position of a hydraulic motor also with application visualizing control process created in software Promotic and InduSoft Web Studio.

At the beginning, it is necessary to describe superficially some functions and features of PAC WinCon. WinCon is industrial controller based on operating system Windows CE .NET and equipped with processor Intel Strong ARM 206 MHz. Connectivity contains Ethernet port, serial RS-232 and RS-485, USB and finally ports VGA and PS/2. Unlike standard PLC it has feature that enables build visualization application on monitor connected straight to the main unit of PAC as well as it can run web server and web pages just in this PAC. Possibilities of design control algorithms are at first by Microsoft Visual Studio and SDK, that can be downloaded from manufacturer's web pages. Software ISaGraf would be another solution but it is relatively expensive and runtime of this software must be already preinstalled in WinCon before buying. Finally, software REX can be used.

Control system REX can be run on many platforms, for example on Windows XP/Vista and Windows CE .NET. When it is run on PAC WinCon, it can be used for deterministic real-time control of many processes. There are supported larger half of modules for WinCon, as well as communication standards ModBus and OPC. Through integrated OPC server could be connected almost every application for visualization of processes. Few weeks ago, it has been released also version of REX for WinPAC, that is quite similar to WinCon but with faster CPU, newer operating system and some differences in communication ports.

System REX consists of few components. At first, tool RexDraw is used for graphical design of control algorithms that result into .mdl files compatible with Matlab-Simulink. Tool RexView is a diagnostic tool and allows the user change parameters of blocks or view trends in REX Runtime. RexComp is a tool for compiling .mdl schemes into .rex files that can be through RexView uploaded into RexCore. RexCore run on the target device as a main part of REX and it is responsible for executing .rex files. It also communicates through integrated OPC server that allows access to inputs, outputs and parameters of all blocks in the schemes executed by REX.

Design of control algorithms (.mdl schemes) can be done either in RexDraw or Matlab-Simulink. As a part of installation process, it adds a library called REXLib into libraries of Simulink. For creating scheme, that is possible to compile and upload into RexCore, it must be used only blocks from REX Library. Just for simulating in Simulink can be combined blocks from REX Library and all blocks from usual Simulink Library. In this case, the "Fixed step" must be set up in parameters of simulations.

Every application for REX consists at least of two schemes .mdl. Main scheme contains just block "EXEC", which is used for configuration of parameters like the type of the target device and information for scheduler like periods and priorities of a tasks that can be executed with a different periods and priorities. Further, blocks of drivers are connected to block "EXEC" as well as blocks representing .mdl schemes that will be executed in period and priority set previously. Typical period of executing .mdl scheme in RexCore is about 5 milliseconds, for simpler process this period can be set on 2 milliseconds. Then there are schemes with control algorithm, whose are executed periodically.



In next phase of thesis, it will be described the real usage of REX system in the application controlling position of hydraulic motor. This model application consists of hydro generator that made system pressure, which was through 4-way servo valve regulated into two contra-motion chambers of piston rod. Technological outputs regulated through analog output of module I-87024 were a position of the servo-valve was and a rotation speed of the motor generating system pressure. As inputs, there were used two modules I-8017 for: feedback of a position of the hydraulic motor, system pressure, pressure in both chambers and real position of the servo-valve and real rotating speed of the motor. Exact type of PAC WinCon was W-8731 with seven slots for serial or parallel modules.

The application consist of REX controls scheme at first, then it was designed visualization on a PC in software Promotic and consequently visualization in InduSoft Web Studio in the WinCon. All of this parts will be now described.

Control scheme in REX contained as main element PID regulator “PIDMA” with integrated auto-tuner based on the method of moments by REX Controls. It enables control in automatic mode, when the position is controlled through closed loop and in manual mode, when the system works in open loop. In automatic mode, system allowed manual order of the position or automatically generated sequence of the positions. All technologic data were being stored into trend buffers and can be read afterwards in RexView. Number of trended technological variables was eight, sample time of periodic archiving these variables was the same as the period of executing this control scheme.

The period of executing this scheme was set on 10 milliseconds. According to RexView, there were diagnosed these times: average time of computing one cycle as 1.38 milliseconds and max time cycle as 3.4 milliseconds. Then the application in InduSoft Web Studio was run and behavior of both programs was observed. System REX continued in running the same way as before and visualization in InduSoft run also steadily. Because these results showed that this setting is not on the limit of the PAC, the period of executing control scheme has been changed to 4 milliseconds. This period led to same results in REX as well as the maximal time cycle wasn't exceed. Differences were detected in behavior of InduSoft because some operations were measurably slowed-down. This showed that priorities of tasks in REX are much higher than priority of visualization. So this was the limit of control on the device WinCon with REX and InduSoft.

Communication between InduSoft and REX was realized through OPC Server. In this case when target device for InduSoft is different than station where the application is being designed, all the connection settings are more difficult, than in the case of visualization in Promotic. REX running on WinCon has in-process OPC server, which can be accessed only from local device and not from remote devices on Ethernet. This leads to the problem, that during design of application in InduSoft, target OPC server can not be connected and setting up this connection is quite “blindly”. On the other side, there exists a solution with runtime of REX running on PC. Every REX scheme for WinCon can be easily redesign to run on a PC platform, just after deleting few I/O blocks and drivers. This scheme can be uploaded to local RexCore on the PC and because the in-process OPC server exists in same version for PC and WinCon, the connection can be easily set up to local OPC server of REX and after uploading the resulting application to WinCon, this connection will also be established.

With software InduSoft was also designed an application as web pages that were placed into Web-server directory of WinCon and allowed access to control of this process through usual web browser on devices connected to Ethernet.

Another solution is to use software Promotic that is intended to use on PC. Although OPC server in REX can't be reached remotely, there exist a way how to get around this troubles. OPC

server of local RexCore can be set up to listen that OPC server on WinCon, so the connection of visualization application will be established to local OPC server, but practically it will communicate with WinCon.

There was also one difference in speed of refresh screens and especially trends in Promotic and InduSoft. While the Promotic could refresh trend via OPC every 100 milliseconds, the runtime of InduSoft in WinCon has limit at least one second, faster refreshing of trends on screen is not possible. It has been found out similar limits for web-based visualization.

In Matlab-Simulink can be designed scheme that could communicate with RexCore on different device using block “RDC” from REX library. This block sends and receives data from another block with special ID and on a device with specified IP address, so these blocks must be paired. Communication it selves is based on UDP/IP, so no packets sent or received between those two blocks shouldn't be lost. Every block RDC has 16 inputs and 16 outputs. Values on inputs of the first block RDC are transferred into outputs of the second block RDC that is usually placed on a different device.

In this thesis, at first, schemes were build either for WinCon and Simulink. The main scheme with process of control position of hydraulic motor was placed in Simulink, while the scheme for REX in WinCon did not have any other function than collecting technological values of controlled process through analog inputs of WinCon and setting the analog outputs. These values were continuously transferred between the WinCon and the scheme in Simulink. This kind of control is very suitable for design and testing whole control process, because changes can be very easily applied even during execution. When the control algorithm was successfully developed, this algorithm was converted to scheme for REX in WinCon and scheme in Simulink was redesigned just for viewing technological variables through blocks “SCOPE”.

In both cases, schemes in Simulink had to be set of “Fixed step” and specified period. To enable executing scheme periodically block “SLEEP” from REX library had to be added. This block is responsible just for a determined periodical execution of the scheme and specified period must be the same as the sample time in Configuration parameters of Simulink.

Limits of periodical execution scheme with block RDC is about 100 milliseconds. With this settings, the simulation process in Simulink run in real-time. But when the period was chosen lower, computing of whole algorithm gets slower and it didn't respected the time of simulation. It means that simulation in real took longer time than it was set. So the measurement of time with Matlab function “etime” has been done. It shows results, that when setting parameter SLEEP to 80 milliseconds, the time of real execution was 17.5 seconds instead of 15 seconds and even when 20 milliseconds were send, the time of real execution took 23.4 seconds. The results proved that scheme containing blocks RDC can ton be usually executed faster then with 100 milliseconds period.

Then a measurement was done to determine differences between parallel and serial modules for WinCon. It was based on function, which regularly periodically change the analog output voltage of the modules between two levels, which was observed on a digital oscilloscope. The rate of analog output changes was the same as period of executing the main control scheme, so it was set to 10 milliseconds. There were used parallel module I-8024 and serial module I-87024. The results of measurement was conclusion, that module I-8024 is able to change analog output level at period 10 milliseconds, while the module I-87024 changed its output level at minimal period of 30 milliseconds. Behavior of communication between modules, main CPU and control system REX is not described in manual of WinCon, neither in REX. On the other side, measurement also proved the deterministic

real-time characteristics of controls system REX at 10 milliseconds level. Measurement, that consists of 20 samples, showed that average period was 9,845 milliseconds, median 10 milliseconds and standard deviation 0,37 milliseconds.

Control system REX is a relatively new product that is still being developed to add new features or remove known errors. In the time, when the work on the diploma thesis started, the control system exists in version 1.30 while actual version during writing this text is 1.42. In older versions, there weren't described all necessary parameters and settings in the manual, as well as there weren't supported all target platforms as in actual version. For WinCon it means that much more modules are now supported. Also it would be appropriate to mention newly support for PAC from ICPDAS called WinPAC, that compared to WinCon improves computing performance and operation system – MS Windows CE 5.0 and other communication ports.

A design of control algorithms in Simulink is not possible only with REX system and WinCon. ICPDAS provides Compact PAC called iPAC-8000 that exists also with preinstalled runtime providing execution of standard Simulink scheme. Package with this PAC includes drivers and blocks for Simulink, especially block for access to I/O modules. The drivers consist of compiler that enables running the Simulink scheme in this PAC. Compared to PAC WinCon, it has better real-time performance as advantage. Unfortunately only "Discontinues" blocks of the standard Matlab library must be used. Moreover, the device with this control scheme could not communicate with any other device, even OPC standard is not supported, so any visualization of controlled process could not be connected. This results in very close target applications, unlike REX system in WinCon.

The newest PAC from ICPDAS is the XPAC 8000. It is equipped with operation system Windows Embedded Standard 2009, with larger operating and Flash memory. Communication ports are similar to WinPAC. Control algorithms can be designed now only with Visual Studio, but it can be supposed, other development environment will be available soon (for example REX system). Its advantage is, that control algorithm and visualization, both can be designed for example as ASP.NET project in Visual Studio Express, that is free of charge. This web site can be through IIS (IIS is already integrated in this PAC) published in the form of web pages and can be browsed in every web browser on devices in a same net.

As a conclusion, WinCon with system REX can be applied to control of industrial medium-rate machines processes with strict requirements on the sampling period stability. Typical minimum sampling period of executing schemes can be between 5 and 10 milliseconds, what was confirmed by model application controlling position of hydraulic motor. In this case, parallel modules in WinCon must be used, while serial modules don't realize the conditions to period of desired refresh rate.

## 7. Použitá literatura

- [1] Tutsch, M. *Vzdálené řízení a vizualizace procesů na SoftPLC WINCON 8000 s Windows CE .NET*. Ostrava, 2007. 86s. Bakalářská práce na Fakultě elektrotechniky a informatiky VŠB-TUO na katedře měřicí a řídicí techniky. Vedoucí diplomové práce Ing. Ondřej Krejcar.
- [2] Internetové stránky ELVAC a.s. [online]. Dostupné na: <http://www.icpcon.cz>
- [3] Internetové stránky ICPDAS Co.,LTD [online]. Dostupné na: <http://www.icpdas.com>
- [4] Martinásková, M. *Programovací jazyky pro PLC*. Automatizace [online]. Červen 2004. Dostupné na WWW: <<http://www.automatizace.cz/article.php?a=142>>
- [5] Internetové stránky REX Controls s.r.o. [online]. Dostupné na: [www.rexcontrols.cz](http://www.rexcontrols.cz)
- [6] Kačmář, D. *Programujeme .NET aplikace ve Visual Studiu .NET*. ISBN 80-7226-569-5
- [7] Fox, D., Box, J. *Building Solutions with the Microsoft .NET Compact Framework*. 2003. ISBN 0-321-19788-7
- [8] Andy Wigley, Stephen Wheelwright. *Microsoft .NET Compact Framework*. ISBN 0-7356-1725-2
- [9] Boling, D. *Programming Microsoft Windows CE .NET*. 2003. ISBN 0-7356-1884-4
- [10] Lacko, I. *Programujeme mobilní aplikace ve Visual Studiu .NET*. ISBN 80-251-0176-2
- [11] Internetové stránky ICE Technology [online].  
Dostupné na: <http://www.icetechnology.com/>
- [12] Internetové stránky Mathworks [online]. Dostupné na: <http://www.mathworks.com/>
- [13] *Řídicí systém REX – Uživatelská příručka*. REX Controls s.r.o., Plzeň, 2001.
- [14] *Funkční bloky systému REX*. REX Controls s.r.o., 4. vydání, Plzeň, 2004.
- [15] Internetové stránky ICPDAS – USA Co.,LTD [online]. Dostupné na:  
<http://www.icpdas-usa.com/>
- [16] Noskievič, P. *Modelování a identifikace systémů*. 1. vydání. Ostrava: Montanex a.s., 1999, ISBN 80-7225-030-2
- [17] Schlegel, M., Balda, P., Štětina M. *Robustní autotuner – momentová metoda*. Automatizace. 2003, roč.46, č.4, s.242-246, ISSN 0005-125X.

## **8. Seznam příloh**

Příloha I - Náhrady jednotlivých bloků z knihovny v prostředí Matlab-Simulink bloky podporovanými systémem REX

Příloha II - Měření reálné periody spouštění řídicího schématu REXu

Příloha III - Struktura a obsah přiloženého CD

**Příloha I - Náhrady jednotlivých bloků z knihovny v prostředí Matlab-Simulink bloky podporovanými systémem REX**

Následuje přehled možných náhrad jednotlivých bloků z knihovny v prostředí Matlab-Simulink bloky podporovanými systémem REX (z REX Library). Jednotlivé bloky Simulinku jsou obsaženy v souboru „simulink.mdl“ a „simulink\_extras.mdl“ v adresáři MATLAB/R2007b/toolbox/simulink/blocks. Otevřením tohoto souboru v Matlabu se zobrazí obsah této knihovny a jednotlivé rozčlenění – „kategorie“ bloků. Každou „kategorii“ je možné dvojitým poklepnutím znovu otevřít a zobrazit jednotlivé bloky, které k ní náleží. Popis jednotlivých bloků je po řadě tak, jak jsou v těchto kategoriích zařazeny. V tabulce je vždy schéma jednotlivých bloků ze Simulinku a jejich možných náhrad z knihovny REXu.

Použitý software:

Matlab verze 2007b (The MathWorks)

REX verze 1.41 (REX Controls)

## Přehled popsaných knihoven ze Simulinku:




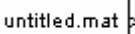
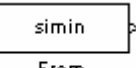
<b>LIBRARY: SIMULINK.....</b>	<b>3</b>
SOURCES.....	3
<i>Model &amp; Subsystem Inputs .....</i>	<i>3</i>
<i>Signal Generators.....</i>	<i>3</i>
SINKS.....	4
<i>Model &amp; Subsystem Output .....</i>	<i>4</i>
<i>Data Viewers .....</i>	<i>5</i>
<i>Simulation Control.....</i>	<i>5</i>
CONTINUOUS.....	5
<i>Continuous-Time Linear System.....</i>	<i>5</i>
<i>Continuous-Time Delays .....</i>	<i>6</i>
DISCRETE.....	6
<i>Discrete-Time Linear System.....</i>	<i>6</i>
<i>Sample &amp; Hold Delays .....</i>	<i>7</i>
DISCONTINUITIES.....	7
SIGNAL ROUTING.....	8
<i>Signal Routing .....</i>	<i>8</i>
<i>Signal Storage &amp; Acces .....</i>	<i>10</i>
SIGNAL ATTRIBUTES.....	10
<i>Signal Attribute Manipulation .....</i>	<i>10</i>
<i>Signal Attribute Detection .....</i>	<i>10</i>
MATH OPERATIONS .....	11
<i>Math Operations.....</i>	<i>11</i>
<i>Vector/Matrix Operation .....</i>	<i>12</i>
<i>Komplex Vector Conversions.....</i>	<i>12</i>
LOGIC AND BIT OPERATIONS .....	12
<i>Logic Operations .....</i>	<i>12</i>
<i>Bit Operations.....</i>	<i>13</i>
<i>Edge Detection .....</i>	<i>13</i>
LOOKUP TABLES.....	13
USER-DEFINED FUNCTIONS.....	14
MODEL VERIFICATION .....	15
PORTS & SUBSYSTEMS .....	15
MODEL-WIDE UTILITIES.....	15
<i>Linearization of Running Models .....</i>	<i>15</i>
<i>Documentation.....</i>	<i>15</i>
<i>Modeling Guides.....</i>	<i>15</i>
<b>LIBRARY: SIMULINK EXTRA.....</b>	<b>16</b>
ADDITIONAL LINEAR .....	16

## Library: Simulink





















Knihovna standardních bloků Simulinku.

### Sources











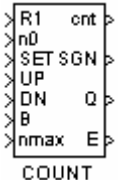

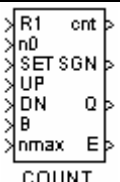
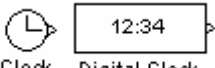
#### Model & Subsystem Inputs

Simulink	REX	Pozn.
 In1	 In1	
 Ground  untitled.mat From File  simin From Workspace	Bloky nemají v REXu alternativu.	

### Signal Generators



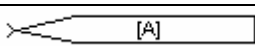
Simulink	REX	Pozn.
 Constant	 CNR  CNI  CNB	Konstanta reálná, celočíselná a bitová.
 Signal Generator	 SG  RUN SYN SGI	SG,SGI – generátory periodického harmonického, obdélníkového, pilovitého signálu případně šumu.
 Pulse Generator	 RUN y is ANLS	
 Signal Builder	 RUN y is ANLS	
 Ramp	 u1 u2 y SW SWR	Sice trochu odlišný blok ale umožňuje tuto funkci (rampu) vytvořit.
 Sine Wave	 SG  RUN SYN SGI	
 Step	 RUN y is ANLS	
 Repeating Sequence  Chirp Signal	Bloky nemají v REXu alternativu.	



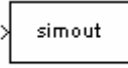


 Random Number	 FNX	
 Uniform Random Number	 FNX	
 Band-Limited White Noise	 SG SGI	
 Repeating Sequence Stair	 ANLS	Blokem „ANLS“ lze nadefinovat krátkou posloupnost impulsů (definovanou čtyřmi uzlovými body).
 Repeating Sequence Interpolated	Blok nemá v REXu alternativu.	
 Counter Free-Running	 COUNT	
 Counter Limited	 COUNT	
 Clock Digital Clock	Bloky nemají v REXu alternativu.	




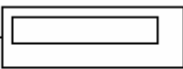
## Sinks

### Model & Subsystem Output


Simulink	REX	Pozn.
 Out1	 Out1	Výstupní port (Outport) slouží k propojování signálů mezi jednotlivými úrovněmi hierarchie.
	 [A]	Připojení výstupního signálu z řídicího algoritmu.

 Terminator  untitled.mat To File  simout To Workspace	Bloky nemají v REXu alternativu.	
---	----------------------------------	--

## Data Viewers


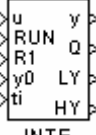


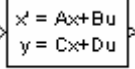
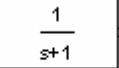
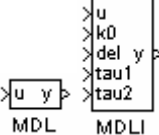
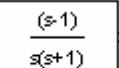
Simulink	REX	Pozn.
 Scope  Floating Scope  XY Graph  Display	Bloky nemají v REXu alternativu.	S řídicím systémem REX lze komunikovat přes OPC nebo přes Simulink s blokem „RDC“ a tato naměřená data následně vizualizovat.

## Simulation Control


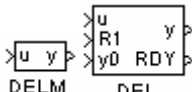

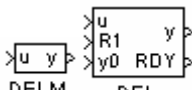

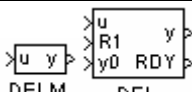
Simulink	REX	Pozn.
 STOP Stop Simulation	Blok nemá v REXu alternativu.	Pozastavení běžícího algoritmu v REXu lze provádět z nástroje RexView.

## Continuous

### Continuous-Time Linear System


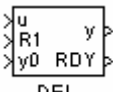

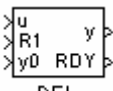




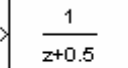
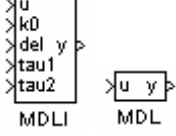
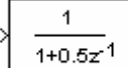


Simulink	REX	Pozn.
 Integrator	 INTE	
 Derivative	 DER	
 State-Space	Blok nemá v REXu alternativu.	
 Transfer Fcn	 MDL MDLI	Pro model soustavy se dvěma časovými konstantami. Parametry soustavy lze u bloku „MDLI“ měnit za chodu ze vstupů bloku.
 Zero-Pole	Blok nemá v REXu alternativu.	

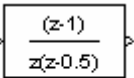
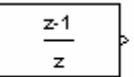

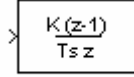
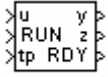
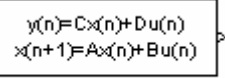
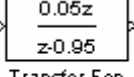
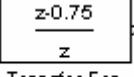

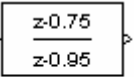


## Continuous-Time Delays

Simulink	REX	Pozn.
 Transport Delay		
 Variable Time Delay		
 Variable Transport Delay		

## Discrete

### Discrete-Time Linear System







Simulink	REX	Pozn.
 Unit Delay		
 Integer Delay		zpoždění o n vzorků
		zpoždění o časový úsek
 Tapped Delay	Blok nemá v REXu alternativu.	
 Discrete-Time Integrator		
 Discrete Transfer Fcn		Pro model soustavy se dvěma časovými konstantami. Parametry soustavy lze u bloku „MDLI“ měnit za chodu ze vstupů bloku.
 Discrete Filter		dolní propust
		pásmová propust

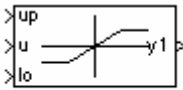

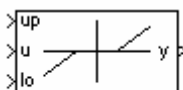








 Discrete Zero-Pole	Blok nemá v REXu alternativu.	
 Difference	 DIF_	
 Discrete Derivative	 DER	
 Discrete State-Space	Blok nemá v REXu alternativu.	
 Transfer Fcn First Order  Transfer Fcn Real Zero	 MDL	
 Transfer Fcn Lead or Lag	 MDL	
 Weighted Moving Average	Blok nemá v REXu alternativu.	

## Sample & Hold Delays

Tyto bloky nemají v REXu alternativu.



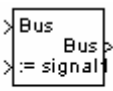









## Discontinuities

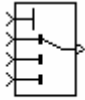
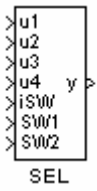




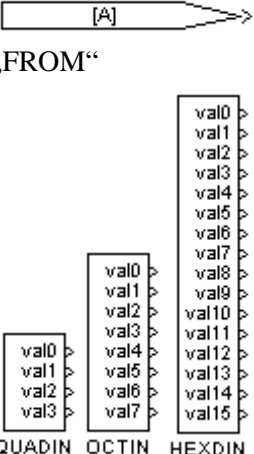

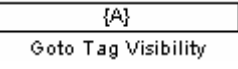

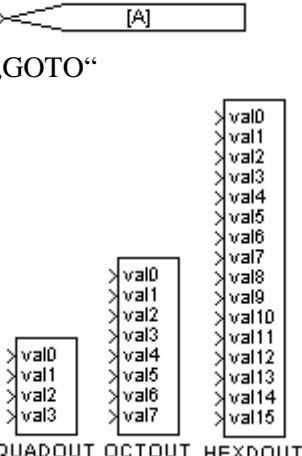
Simulink	REX	Pozn.
 Saturation	 SAT	
 Dead Zone	 LIN	
 Rate Limiter	 RLIM	

 <p>Saturation Dynamic</p>	 <p>SAT</p>	
 <p>Dead Zone Dynamic</p>  <p>Rate Limiter Dynamic</p>	Bloky nemají v REXu alternativu.	
 <p>Backlash</p>  <p>Relay</p>	 <p>RLY</p>	
 <p>Quantizer</p>  <p>Hit Crossing</p>  <p>Coulomb &amp; Viscous Friction</p>  <p>Wrap To Zero</p>	Bloky nemají v REXu alternativu.	


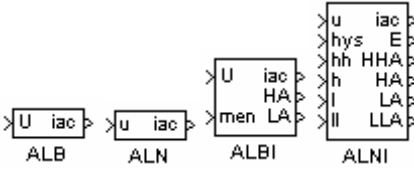
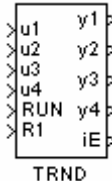
## Signal Routing

### Signal Routing

Simulink	REX	Pozn.
 <p>Bus Creator</p>  <p>Bus Selector</p>  <p>Bus Assignment</p>  <p>Mux</p>  <p>Demux</p>	Bloky nemají v REXu alternativu.	Vytvořit „sběrnici“ v REXu nelze. Multiplexovat v REXu nelze.
 <p>Selector</p>  <p>Index Vector</p>  <p>Merge</p>  <p>Environment Controller</p>	Neexistuje ekvivalent (provádí výběr prvku z matice nebo vektoru a ty nejsou v REXu podporovány)	
 <p>Manual Switch</p>	 <p>SSW</p>	Pro proměnné typu „double“.
	 <p>ISSW</p>	Pro proměnné typu „integer“.



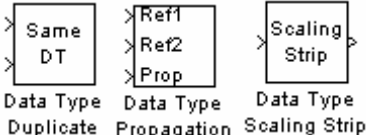


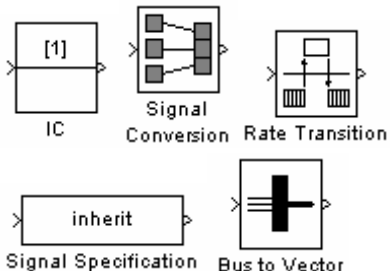
 <p>Multiport Switch</p>	 <p>SEL</p>	
 <p>Switch</p>	 <p>SSW</p>	Pro proměnné typu „double“.
	 <p>ISSW</p>	Pro proměnné typu „integer“.
 <p>From</p>	 <p>„FROM“</p>	
 <p>Goto Tag Visibility</p>	 <p>Goto Tag Visibility</p>	
 <p>Goto</p>	 <p>„GOTO“</p>	

## Signal Storage & Acces

Simulink	REX	Pozn.
 <p>Data Store Read   Data Store Memory   Data Store Write</p>	Bloky nemají v REXu alternativu.	
	 <p>ALB   ALN   ALBI   ALNI</p>	Alternativní bloky pro zaznamenávání alarmů v REXu
	 <p>TRND</p>	Pro ukládání trendů v REXu slouží blok „TRND“.

## Signal Attributes

### Signal Attribute Manipulation


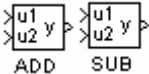

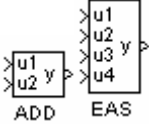

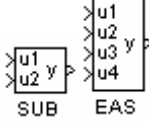
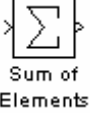







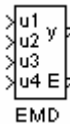


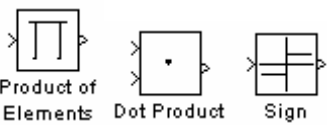




Simulink	REX	Pozn.
 <p>Convert</p> <p>Data Type Conversion</p>	 <p>RTOI</p>	Blok „RTOI“: převod reálné konstanty na celočíselnou
 <p>Same DT   Data Type Duplicate   Ref1   Data Type Propagation   Ref2   Data Type Scaling Strip   Prop</p>	Bloky nemají v REXu alternativu.	
 <p>Convert y</p> <p>Data Type Conversion Inherited</p>	 <p>RTOI</p>	Blok REXu: převod reálné konstanty na celočíselnou
 <p>IC   Signal Conversion   Rate Transition   inherit   Signal Specification   Bus to Vector</p>	Bloky nemají v REXu alternativu.	Inicializace hodnoty signálu při spuštění simulace.

## Signal Attribute Detection



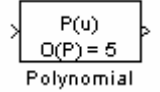


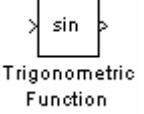

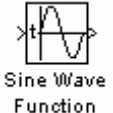
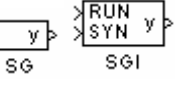
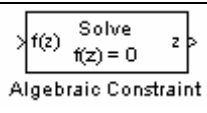
Tyto bloky nemají v REXu alternativu.

## Math Operations

### Math Operations

Simulink	REX	Pozn.
 Sum		
 Add		Blok „EAS“ pro více vstupních signálů
 Subtract		Blok „EAS“ pro více vstupních signálů
 Sum of Elements	Blok nemá v REXu alternativu.	V REXu na jednom „drátě“ pouze jedna proměnné (nevektorová, nematicová)
 Bias		Offset lze nahradit blokem „ADD“
 Gain		
 Slider Gain		totéž jako „Gain“ pouze s možností úpravy zesílení za běhu simulace pomocí „slideru“ ve vlastním okně
 Product		násobení a dělení vstupních signálů
 Divide		
 Product of Elements   Dot Product   Sign	Bloky nemají v REXu alternativu.	
 Abs		
 Unary Minus		není přímá alternativa, lze obejít přes „Gain“ s parametrem „-1“



 Math Function	 FNX	
 Polynomial	 POL	
 MinMax Running Resettable	Blok nemá v REXu alternativu.	
 Trigonometric Function	 FNX	
 Sine Wave Function	 SG SGI	
 Algebraic Constraint	Blok nemá v REXu alternativu.	

## Vector/Matrix Operation





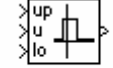
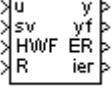

Vektory a matice nejsou v exekutivě REXu podporovány, proto neexistují ani náhrady těchto bloků.





## Komplex Vector Conversions

Komplexní čísla v exekutivě REXu nejsou podporovány, proto neexistují alternativní bloky.



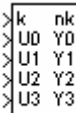



## Logic and Bit Operations

### Logic Operations

Simulink	REX	Pozn.
 Logical Operator	 AND_	
 Relational Operator	 CMP	
 Interval Test Dynamic	 SAI	Blok „SAI“ lze použít mimo jiné i pro zjištění, zda je hodnota proměnné v daném rozsahu.
	 SAT	

 Combinatorial Logic	Blok nemá v REXu alternativu.	
  Compare To Zero    Compare To Constant	 CMP	

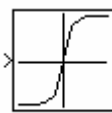

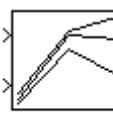
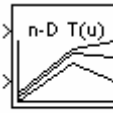
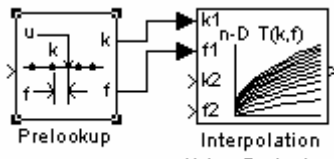
## Bit Operations

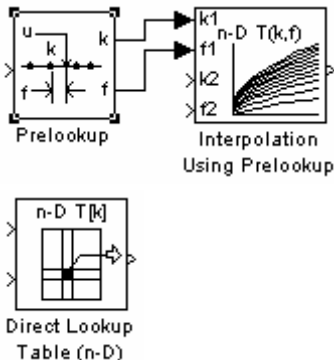
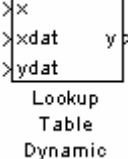

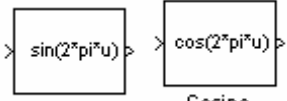

Simulink	REX	Pozn.
  Bit Set    Bit Clear	 ITOI	Blok „ITOI“ pro generování 4-bitového čísla a převod na a z dekadického tvaru.
 Bitwise Operator	  NOT_    OR_	Nutno převést na použití těchto bloků.

## Edge Detection

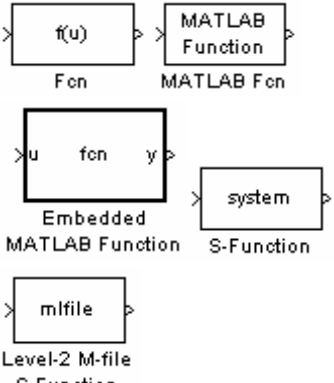
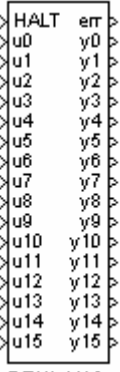
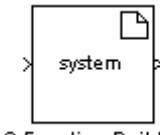
Pro detekci jakýchkoliv náběžných a sestupných hran neexistuje přímo blok, nicméně blok „REXLANG“ lze nakonfigurovat tak, aby tuto funkci zastával.

## Lookup Tables

Simulink	REX	Pozn.
 Lookup Table	 CNDR	
  Lookup Table (2-D)    Lookup Table (n-D)	Bloky nemají v REXu alternativu.	
 Prelookup    Interpolation Using Prelookup	Bloky nemají v REXu alternativu.	

 <p>Prelookup</p> <p>Interpolation Using Prelookup</p> <p>Direct Lookup Table (n-D)</p>		
 <p>Lookup Table Dynamic</p>	 <p>CNDR s blokem SGSPL</p>	<p>Blok SGSPL („speclib“) slouží pro nastavování konstant ostatních bloků.</p>
 <p>Sine</p> <p>Cosine</p>	 <p>FNX</p>	<p>Nelze nastavit rozlišení kroků (počet bitů).</p>





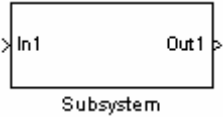
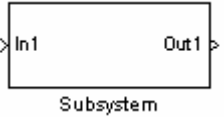
## User-defined Functions

Simulink	REX	Pozn.
 <p>Fcn</p> <p>MATLAB Fcn</p> <p>Embedded MATLAB Function</p> <p>S-Function</p> <p>Level-2 M-file S-Function</p>	 <p>REXLANG</p>	<p>Blok „REXLANG“ umožňuje vytvořit libovolný algoritmus ve skriptovacím jazyce podobném jazyce C.</p>
 <p>S-Function Builder</p>	<p>Blok nemá v REXu alternativu.</p>	

## Model Verification

Tyto bloky nemají v REXu použitelnou náhradu.

## Ports & Subsystems

Simulink	REX	Pozn.
		
		
		

Další bloky z knihovny *Ports & Subsystem* nemají v REXu alternativu.

## Model-Wide Utilities

### Linearization of Running Models

Bloky nemají alternativu v REXu.

### Documentation

Bloky nemají alternativu v REXu.

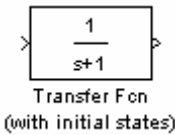
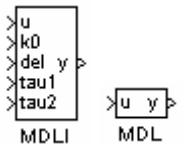
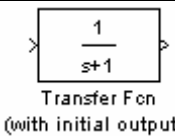
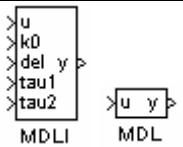
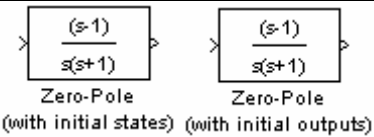
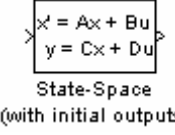
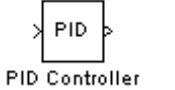
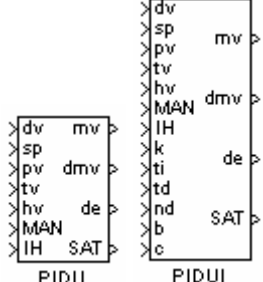
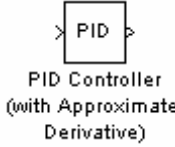
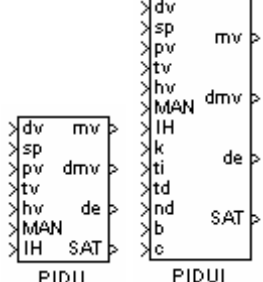
### Modeling Guides

Bloky nemají alternativu v REXu.

## Library: Simulink Extra

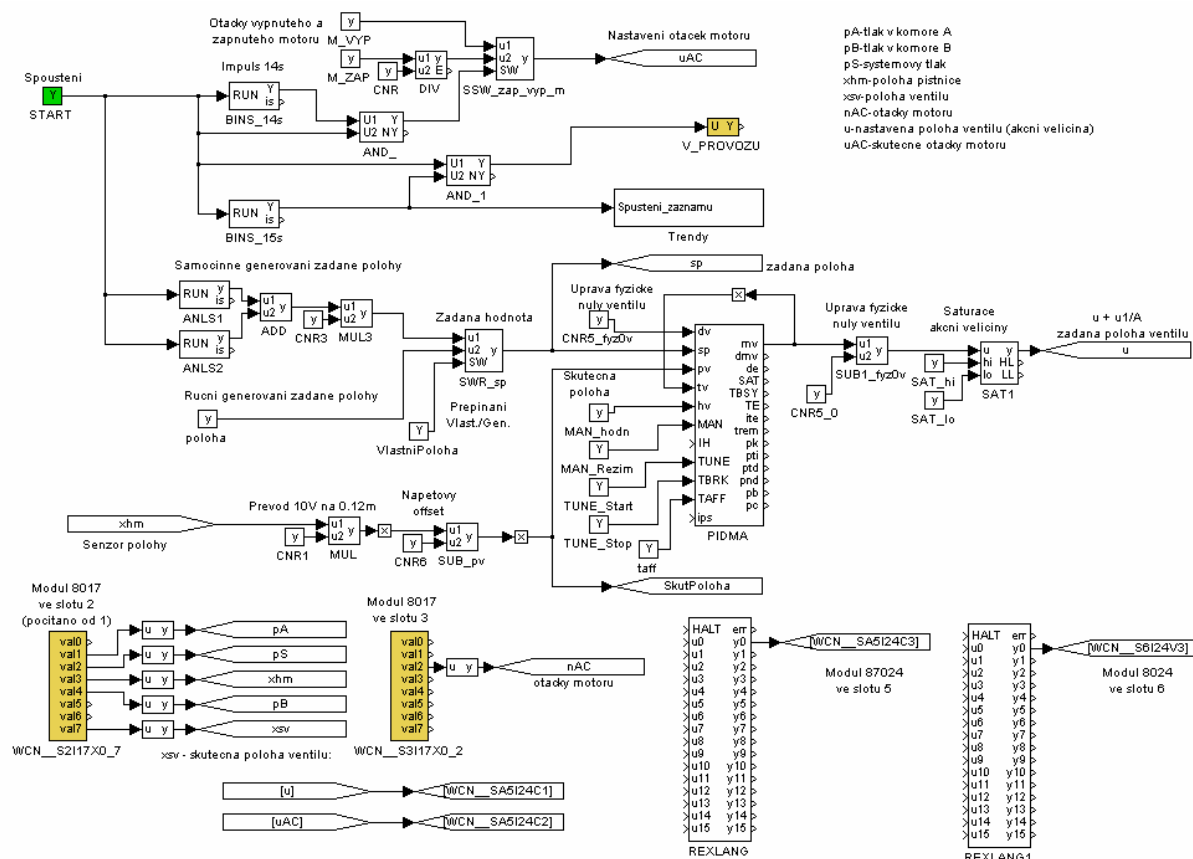
Z této knihovny byla záměrně vybrána jen část „Additional Linear“, která obsahuje bloky PID regulátorů pro spojitou regulaci.

### Additional Linear

Simulink	REX	Pozn.
 <p>Transfer Fcn (with initial states)</p>	 <p>MDLI MDL</p>	
 <p>Transfer Fcn (with initial outputs)</p>	 <p>MDLI MDL</p>	
 <p>Zero-Pole (with initial states) Zero-Pole (with initial outputs)</p>	Bloky nemají v REXu alternativu.	
 <p>State-Space (with initial outputs)</p>	Blok nemá v REXu alternativu.	
 <p>PID Controller</p>	 <p>PIDU PIDUI</p>	
 <p>PID Controller (with Approximate Derivative)</p>	 <p>PIDU PIDUI</p>	

## Příloha II - Měření reálné periody spouštění řídicího schématu REXu

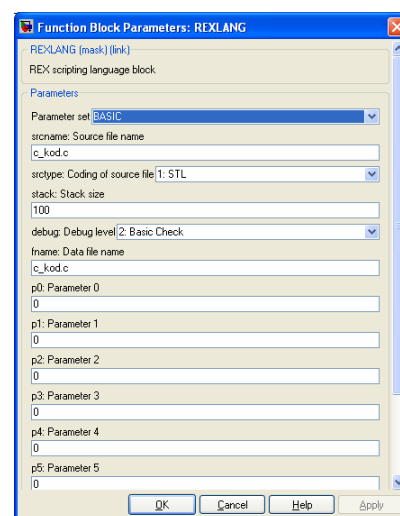
Pro měření real-time odezvy a průměrné periody spouštění schématu REXu bylo vytvořeno řídicí schéma odvozené od schématu „pro připojení různých vizualizačních nástrojů“ – viz Obr. 39.



Obr. 39: Vytvořené schéma pro měření periody schématu REXu

Zdrojový C-kód bloku „REXLANG“:

```
long output(0) vystup;
long count=0;
int init(void){
    return 0;
}
int main(void){
    if(count==0)
        count=1;
    else
        count=0;
    if(count==0)
        vystup=1;
    else
        vystup=2;
}
int exit(void){return 0;}
```

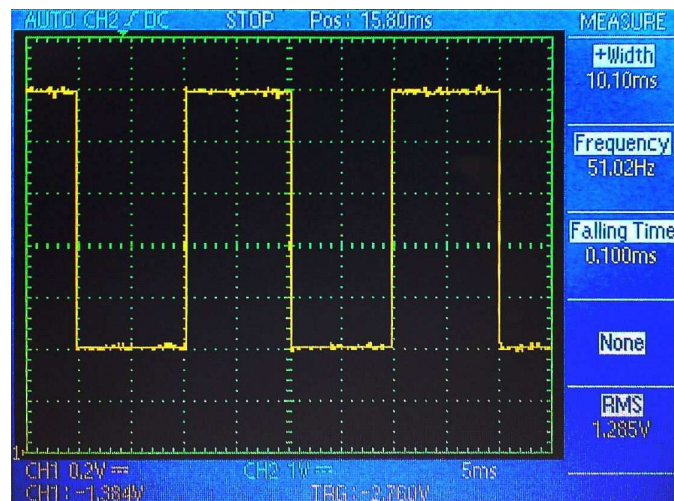


Obr. 40: Nastavení parametrů bloku "REXLANG"

Naměřené časy délky výstupních impulsů analogového modulu I-8024 při periodě spouštění schématu v REXu 10 milisekund a při přepnutí hladiny analogové výstupní hodnoty při každém spuštění schématu:

Číslo měření	Naměřené časy šířky impulsů [ms]
1	10,40
2	9,50
3	10,00
4	10,00
5	10,1
6	10,40
7	9,50
8	10,10
9	9,60
10	10,40
11	9,60
12	10,20
13	9,40
14	10,40
15	9,90
16	10,00
17	10,50
18	9,50
19	9,4
20	10

**Tabulka 12: Naměřená data**



**Obr. 41: Průběh impulsů vytvářený překlápěním obvodu v každém cyklu pro dvě různé výstupní hladiny napětí**

Statistika naměřených dat při spouštění schématu každých 10 milisekund:

Počet vzorků měření	20
Průměr	9,945
Medián	9,830
Rozptyl	0,141
Minimum	9,40
Maximum	10,50
Rozsah	1,1
Směrodatná odchylka	0,37
Standardní chyba	0,08

**Tabulka 13: Statistika naměřených dat**

**Příloha III - Struktura a obsah přiloženého CD**

1. Diplomová práce
2. Aplikace pro vývojové prostředí Simulink, Promotic a InduSoft Web Studio
3. Dokumentace k zařízení WinCon